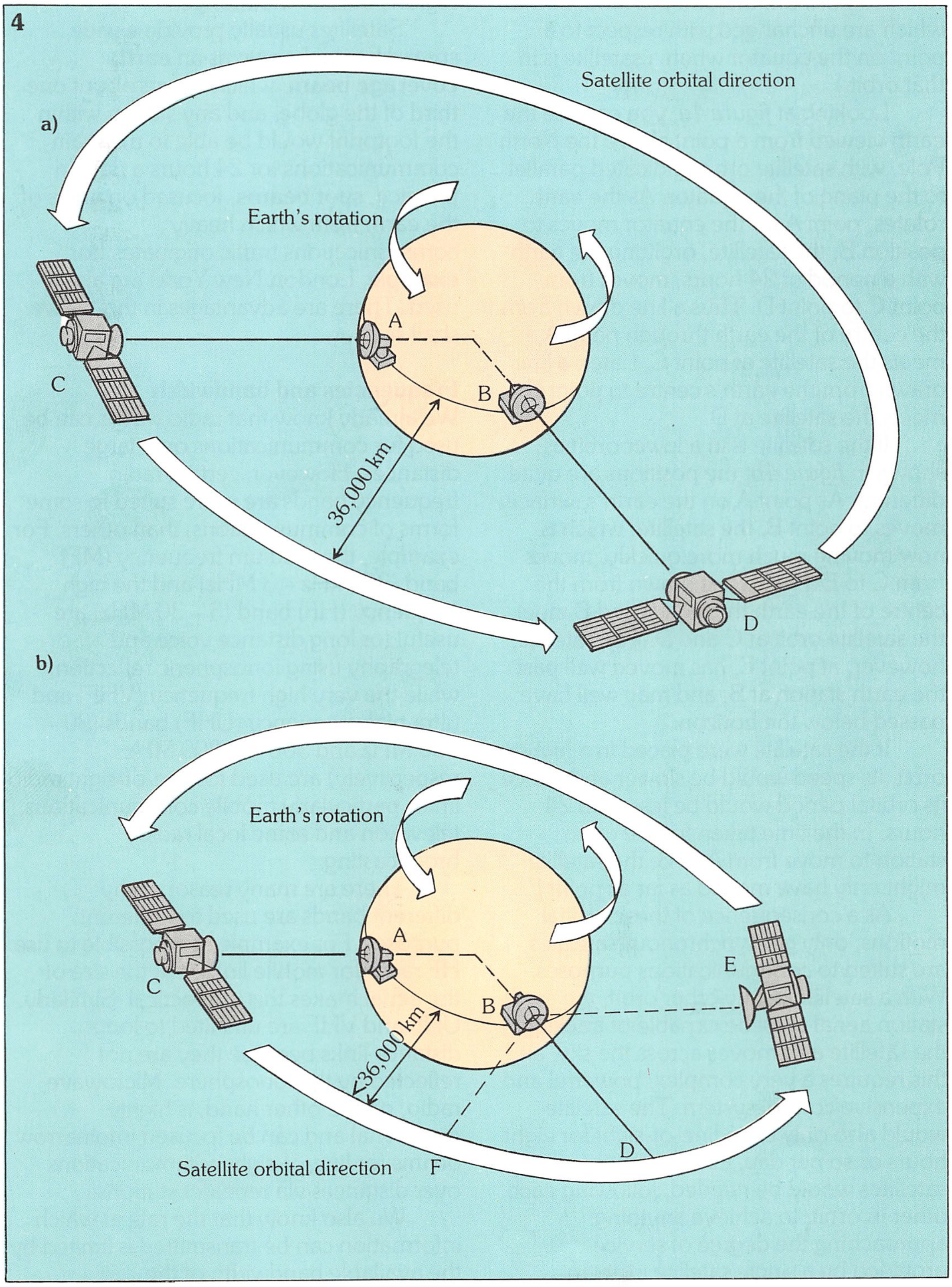
(continued from part 37)

Of orbits and earth

Arthur Clarke's 1945 proposal was a carefully considered and detailed outline of all the major requirements for a satellite communications network that eventually developed according to his predictions.

The orbital height, 36,000 km, and the orbital plane (above the equator) that he proposed were not mere random choices. At that height, a satellite has an orbital period (the time to complete one revolution about the earth) of about 24 hours. If the satellite is positioned above a



4. (a) A satellite in geosynchronous orbit remains at a fixed position relative to a certain reference point on the equator because both are rotating about the earth's axis at the same rate; **(b)** a lower orbit increases the satellite's speed relative to earth.

point on the equator, which also revolves about the earth's polar axis once in 24 hours, the satellite is therefore stationary with respect to that point, because both are rotating about the axis at the same rate.

This orbit is the geostationary or geosynchronous orbit that was mentioned earlier. (The terms are equivalent – one refers to position, the other to time, both of which are unchanged with respect to a point on the equator when a satellite is in that orbit.)

Looking at *figure 4a*, you can see the earth viewed from a point above the North Pole, with satellite orbits indicated parallel to the plane of the equator. As the earth rotates, point A on the equator moves to position B; the satellite, orbitting the earth with a period of 24 hours, moves from point C to point D. Thus a line drawn from the centre of the earth through point A meets the satellite at point C. Later, a line drawn from the earth's centre to point B meets the satellite at D.

If the satellite is in a lower orbit as shown in *figure 4b*, the positions are quite different. As point A on the earth's surface moves to point B, the satellite, which is now moving much more quickly, moves from C to E. Now, lines drawn from the centre of the earth through A and B meet the satellite orbit at C and D. The satellite, however, at point E, has moved well past the earth station at B, and may well have passed below the horizon.

If the satellite were placed in a higher orbit, its speed would be slower and hence its orbital period would be less than 24 hours. In the time taken for the earth station to move from A to B, the satellite might only have moved as far as point F.

As a consequence of these orbital motions, only geosynchronous satellites are suited to communications purposes. With a satellite in any other orbit, the earth station aerial must be capable of tracking the satellite as it moves across the sky, and this requires a very complex, powerful and expensive control system. The satellite would also only be in line-of-sight for eight hours or so per day, and thus many more satellites would be needed, following each other in orbit, to achieve anything approaching the degree of service provided by a single satellite in geosynchronous orbit.

The area of the earth's surface which is in effective line-of-sight communications with a satellite is known as its **footprint**. *Figure 5* illustrates the footprints of three satellites in geosynchronous orbits – between them covering virtually the entire surface of the earth, with only the polar regions outside their range.

Satellites usually provide a wide spread beam, known as an **earth coverage beam**, which covers about one third of the globe, and any station within the footprint would be able to maintain communications for 24 hours a day. In practice, **spot beams**, focused on areas of the earth from which heavy communications traffic originates (for example, London/New York) are also used. There are advantages in this, as we shall see.

Frequencies and bandwidth

We already know that radio waves can be used for communications over large distances. However, certain radio frequency bands are more suited to some forms of communications, than others. For example, the medium frequency (MF) band (300 kHz – 3 MHz) and the high frequency (HF) band (3 – 30 MHz) are useful for long distance voice and telegraphy using ionospheric reflection, while the very high frequency (VHF) and ultra high frequency (UHF) bands (30 – 300 MHz and 300 – 3,000 MHz respectively) are used for line-of-sight radio links, particularly mobile communications, television and some local radio broadcasting.

There are many reasons why different bands are used for different purposes. For example, it is possible to use HF radio for mobile links, but the size of the aerial makes this impractical. Similarly, UHF and VHF are unsuited to long distance links because they are not reflected by the ionosphere. Microwave radio, on the other hand, is highly directional and can be focused into narrow beams for line-of-sight communications over distances via repeater stations.

We also know that the rate at which information can be transmitted is limited by the available bandwidth of the

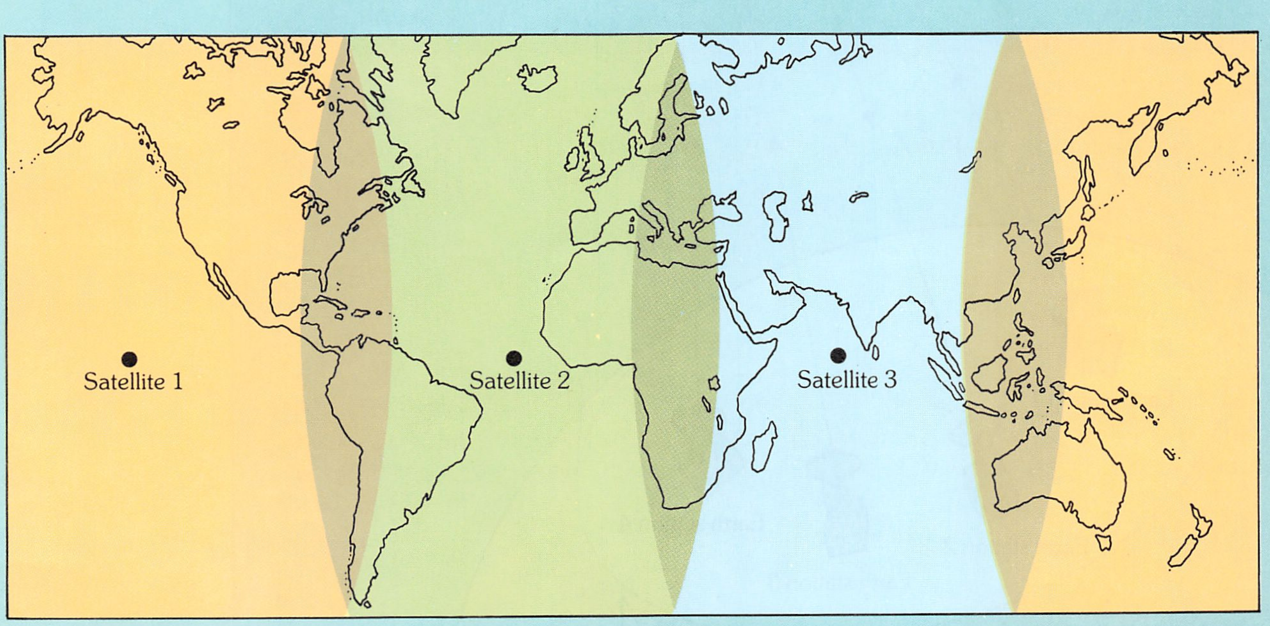
communications link (*Communications* 5 and 6). Thus, to transmit a television program at a rate such that it appears at the receiver as a motion picture, a large bandwidth is needed. (Otherwise the result would be like watching a series of still frames, slowly building up, as with teletext frames which are transmitted at a much slower rate than the picture information.)

We can see, then, that HF or MF bands are totally unsuited for satellite communications because they would generally be reflected from the ionosphere. Earth stations directly below a satellite, with an aerial pointing directly overhead, would be able to penetrate the ionosphere

HF frequencies it would occupy 22% of the available bandwidth between 3 and 30 MHz. Only four and a half television channels could be accommodated, and there would not be any space for the many other communications links that now operate in the HF band. However, at UHF, one channel occupies only 0.22% of the available bandwidth – there is therefore sufficient room for many TV channels, local radio, and the other services currently operating in the UHF region.

Thus, for the communication of large volumes of information (such as facsimile, television, computer data and teletype/ telex services) in reasonable time, and for

5



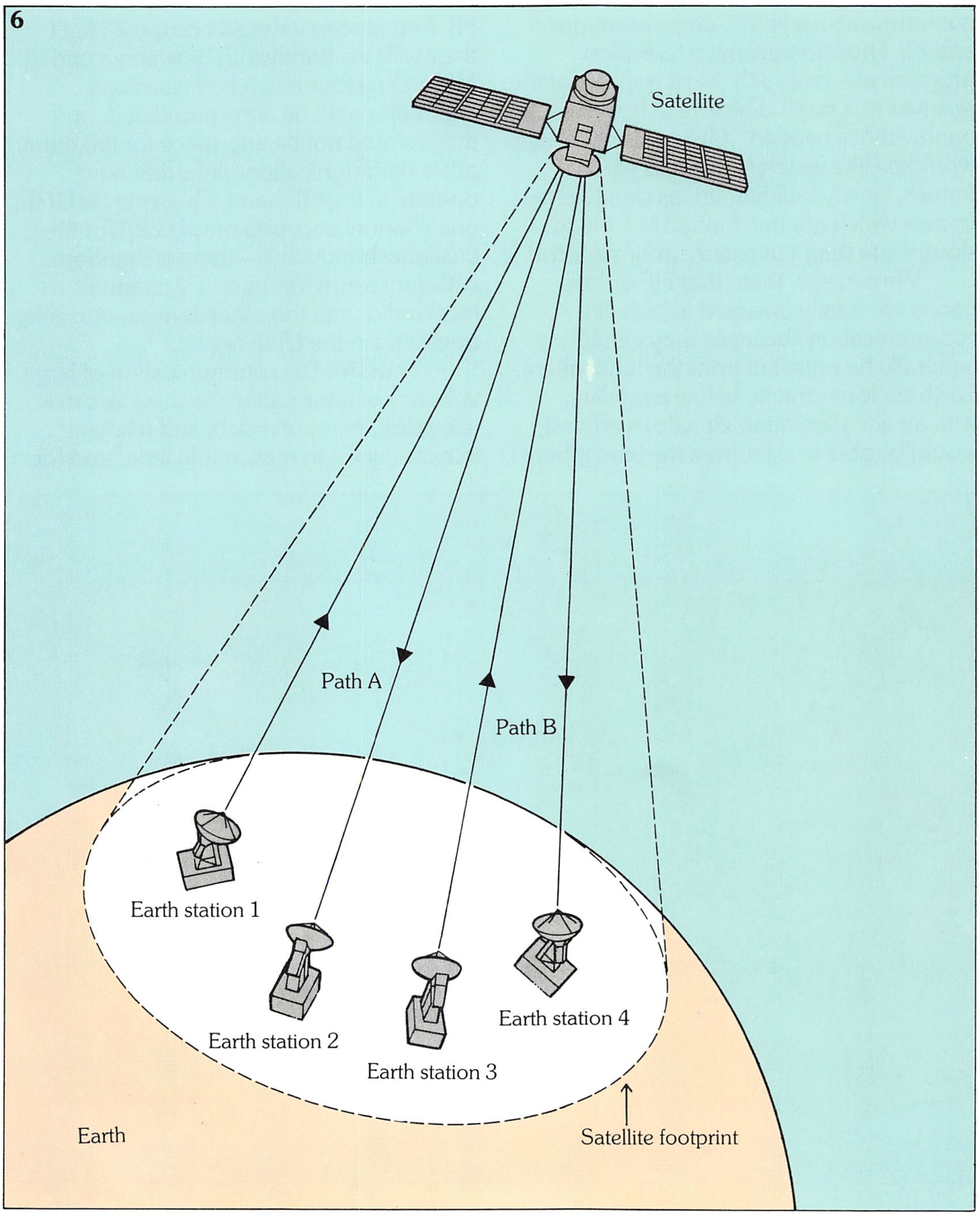
5. The footprints of 3 satellites in geosynchronous orbit overlap slightly, so that each covers about one third of the earth's surface.

with MF or HF waves, but since the majority of earth stations lie north or south of the equator their aerials are inclined to an angle to the vertical, and hence radio waves transmitted from these would not reach the satellite. Further, since the physical size of an aerial is related to the wavelength, very large aerials would be needed both at the earth station and at the satellite.

A reasonable bandwidth for television transmissions, we discovered, is 6 MHz – but if that bandwidth were transmitted at

establishing the links needed to handle many thousands of international telephone calls every day, large amounts of bandwidth are called for. And that means operating at UHF or higher frequencies.

Most satellite communications use frequencies above 1000 MHz (i.e. 1 GHz). The standard nominal frequencies used by most systems now are 6 GHz for the up-link and 4 GHz for the down-link, but the latest satellites, for example INTELSAT V and ECS, use 14 GHz on the up-link and 12 GHz on the down-link; INTELSAT V

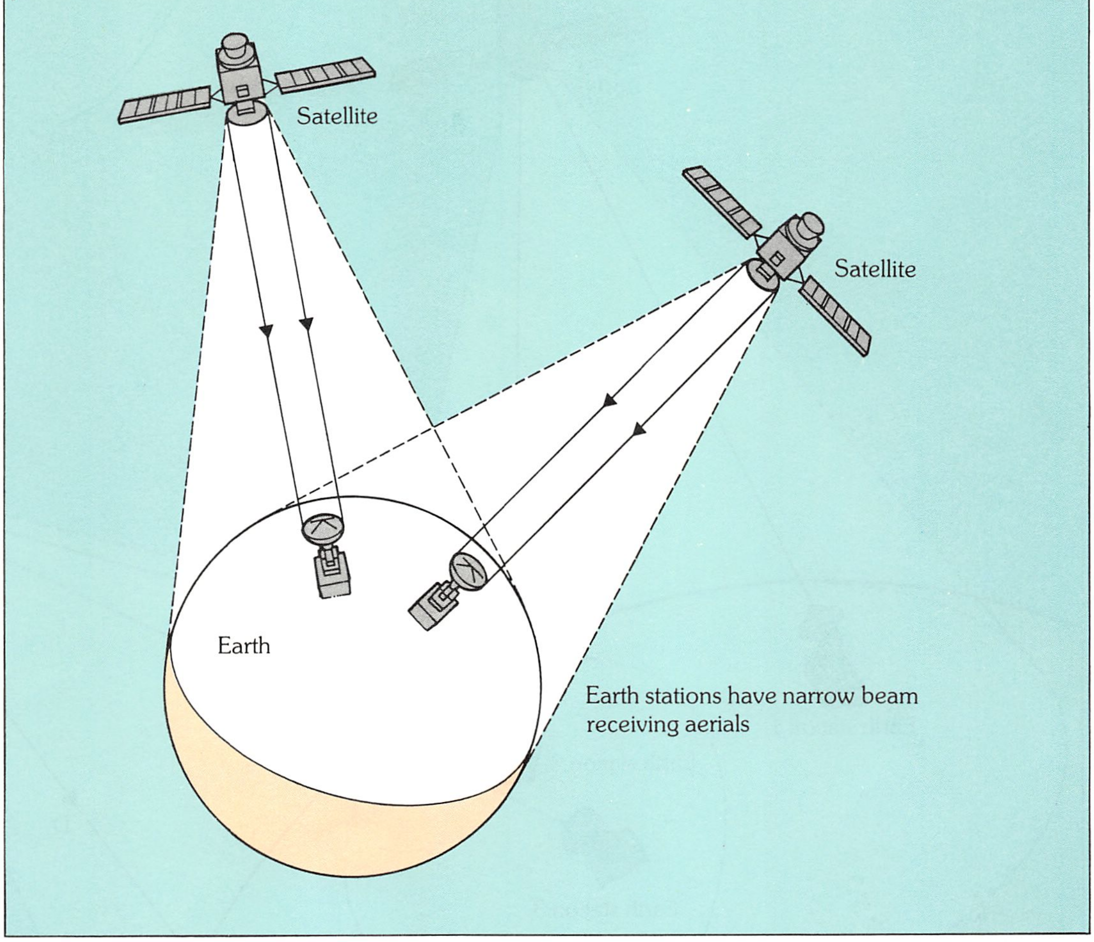


6. Multiple access techniques enable more than one earth station to link with a single satellite.

actually uses both frequency bands. Links of 1.6 GHz and 1.5 GHz are used for maritime ship/satellite links, while other bands are allocated for Direct Broadcast Satellite TV and for military purposes.

The use of different frequencies for up-link and down-link means that full duplex operation is possible, that is, a satellite or earth station can simultaneously transmit and receive. The frequencies are sufficently far apart so as not to cause mutual interference, but close enough to allow a single aerial design to deal with both frequencies. Separate transmit and receive aerials are not needed.

The bandwidths available at these ultra high frequencies are, as we know, very large. Typically, bandwidths of 500 MHz can be achieved, equivalent to 83 television channels. In practice though, not all of this bandwidth can be used because of noise and the need to prevent interference between channels by separating them by a **guard band** of

7


7. Narrow beam receiving aerials are used at earth stations in an attempt to eliminate interference from external sources.

unused frequencies. On the other hand, there are many techniques which, as we shall see, are used to make the most of the bandwidth that is practically available.

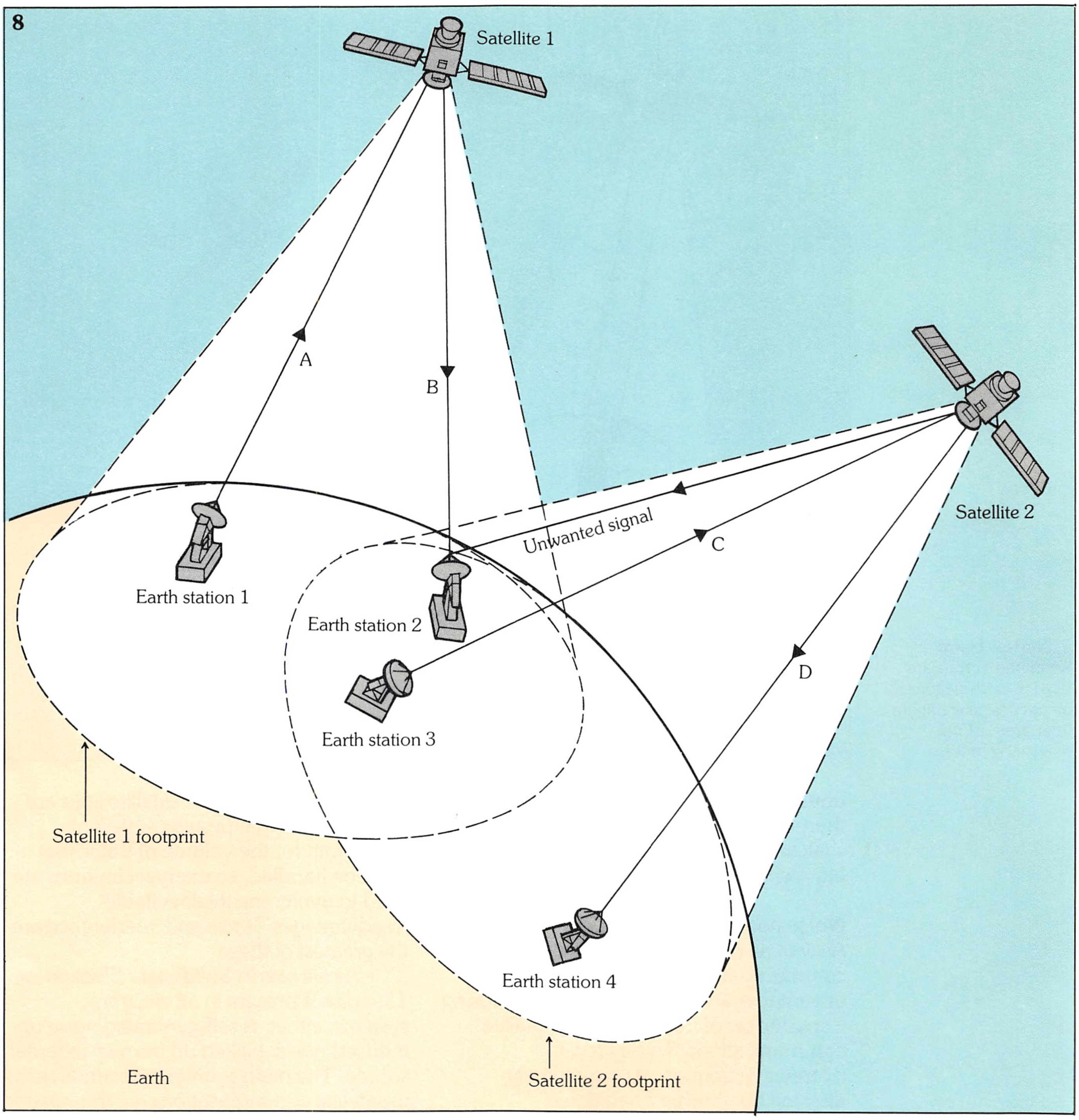
Noise and interference

As with all choices in electronics and communications, there are both advantages and disadvantages in choosing a specific set of frequencies for satellite communications. To begin with, microwave frequencies are severely affected by weather conditions. Furthermore, it is difficult to generate high power transmissions at the satellite, where, typically, 10 W is needed for point-to-point links to earth stations using directional satellite antennae, and up to 1 kW is needed for earth coverage transmissions via satellite.

As a result, there are more problems to be solved at the earth station in separating low level satellite signals from relatively high noise levels. However, the frequencies selected for satellite links are the only ones that provide sufficient bandwidth for the volume of traffic that must be handled, so many techniques are used to overcome the inevitable disadvantages. Noise and interference are the greatest of these.

As we saw in *Solid State Electronics 15*, noise is present in all electronic systems, either as self-generated noise or induced noise, picked up from an external source. The main sources of external noise in satellite communications are the sun and the stars for the earth station, and for the satellite, the earth itself. External noise is overcome by transmitting at the highest possible power levels and by the use of high gain, spot beam aerials wherever practicable.

Internal noise is a major problem, even with digital communications, and to overcome this special **low noise amplifiers** (LNAs) must be used. We will return to look at these, and the **high power**



8. Even with the best aerials available, it is not yet possible to direct an aerial at only one satellite. Interference from unwanted signals from other satellites therefore presents a problem.

amplifiers (HPAs) in the transmitters, in *Communications 8*.

The main factor influencing noise pick-up at the receiver is the aerial beam width. At a given frequency, the larger the diameter of the aerial, the narrower the beam, and at an earth station it is possible to build such large, narrow beam aerials to focus closely on the satellite, eliminating noise pick-up from the sun and stars as far as possible. Unfortunately, large aerials cannot be put into orbit, and so satellite beams cannot be as tightly focused.

But as well as noise from external and internal sources, affecting both the up-link and the down-link, there is another important source of noise.

Until now we have spoken generally of earth coverage and narrow or spot beam links, but we have considered only

two-way communications between two earth stations via a single satellite. Yet this would be highly inefficient because we know that, in theory, any earth station within the footprint of a satellite aerial (be it earth coverage or spot beam) can communicate with any other station in the area. This is an advantage of satellite communications over line-of-sight links, and it ought to be exploited.

In fact we know that there are many satellites in geosynchronous orbit, and that there are over 200 earth stations in the INTELSAT network alone, so it is indeed possible for more than one earth station to link to one satellite. This is achieved by utilising **multiple access** techniques (*figure 6*).

It is also possible for one earth station to link with more than one satellite, but there are problems involved. In order to track several satellites at once, an earth station must use a wide beam receiving aerial, thus increasing the noise pick-up from external sources. This is undesirable so, as far as possible, earth station aerials are directed at one satellite at a time (*figure 7*), though of course they can be repositioned to track another satellite if necessary.

Yet the growth of satellite communications and the increasing demand for links has been so great that with the best aerials now available it is not possible to direct an aerial at one satellite alone, because of the number competing for geosynchronous orbital positions. This is particularly the case over the busy transatlantic route.

As a result, all real satellite communications systems suffer from another form of noise, that is, interference from unwanted signals from other satellites – *figure 8* illustrates the problem. Earth station 1 is transmitting to earth station 2 via satellite 1; but at the same time earth station 3 is transmitting to earth station 4 via satellite 2. Earth station 2, therefore, picks up signals from both satellites and the unwanted signal from satellite 2 is effectively noise, tending to block out the desired signal from satellite 1.

Signals buried in noise (internal, external or interference from another transmission) require far more bandwidth than would otherwise be necessary, and so less efficient use is made of the available bandwidth. Multiple access methods are designed to allow simultaneous communications between a number of satellites and earth stations, whilst making the best use of all of the available bandwidth.

Glossary

down-link	the communications path from an earth station to a ground station, operated on 4 GHz or 12 GHz
footprint	the area of the earth's surface which is in effective line-of-sight communications with a satellite
geosynchronous or geostationary orbit	an orbit which places a satellite at a height of 36,000 km above a point on the equator. A satellite in this orbit remains stationary with respect to the earth
guard band	a band of unused frequencies between channels, left vacant to minimise cross-channel interference
spot beam	a highly directional radio beam from a satellite, throwing a small localised footprint
transponder	satellite communications equipment for down-converting the up-link signal to the frequency band used for the down-link
up-link	the communications path from an earth station to a satellite, usually operated on 6 GHz, but also on 14 GHz with the most recent satellites

2

MICROPROCESSORS

Digital signals in microprocessors

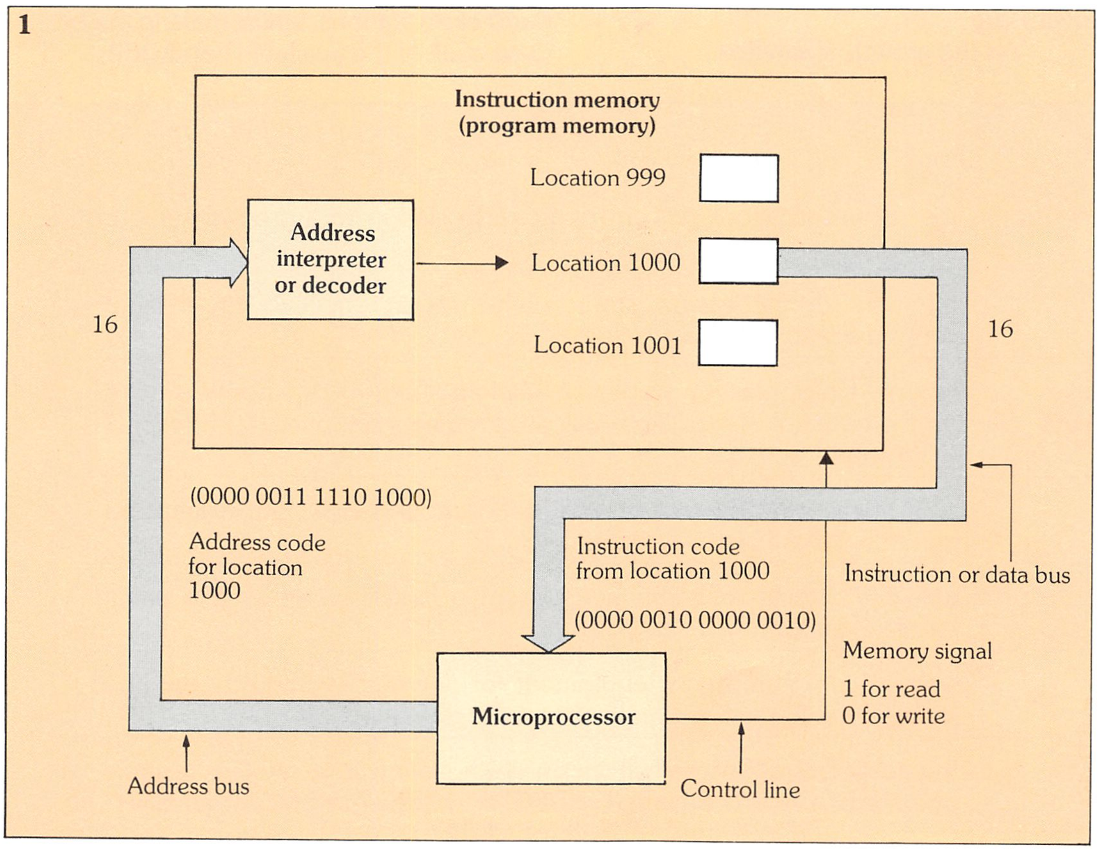
Introduction

As with all digital systems, microcomputers and microprocessors depend on the correctly synchronised transmission and reception of coded electrical signals. In earlier *Digital Electronics* chapters we looked at the different methods by which digital signals can be encoded and manipulated. These principles are employed in microprocessor circuits, but we need to look at them in more detail to be able to fully understand a microprocessor based computer system.

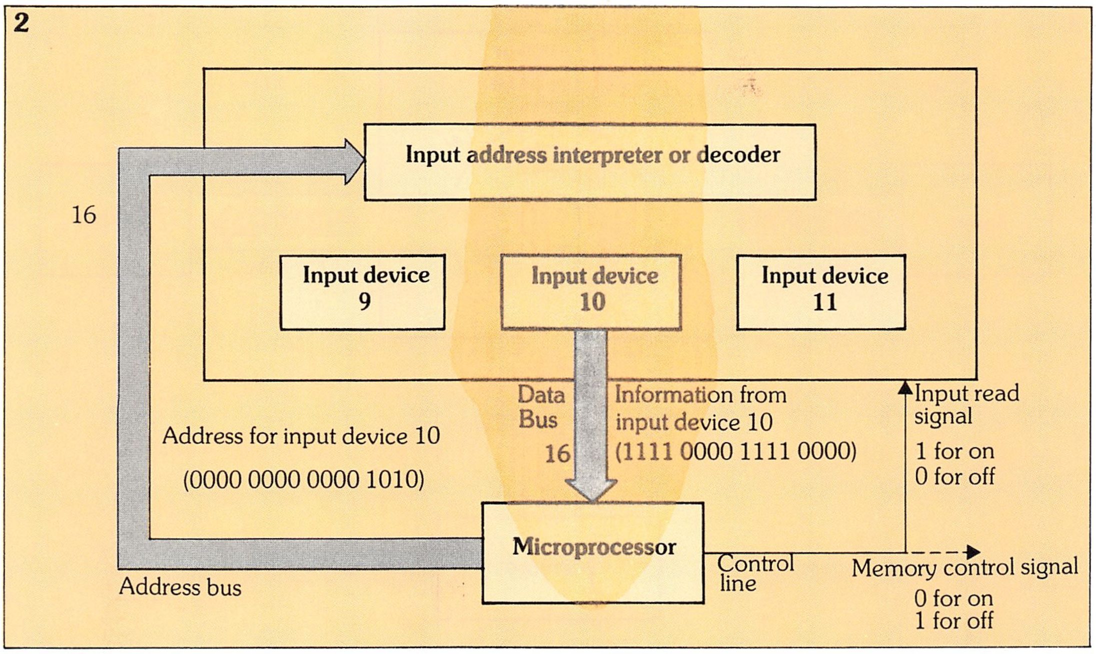
Binary numbers, 4, 8 or 16-bit depending on the microprocessor, can be used to represent numbers, characters, commands, memory addresses and time. Digital signals can mean different things to different subsystem functional blocks, as we saw in the last chapter. Each subsystem must provide the circuitry to decipher the coded binary digits and interpret their meaning.

To recap, let us look at instruction codes, address codes and number and character data codes in a little more detail. **Instruction codes** are the binary codes that correspond to items in the microprocessor's instruction set. For instance, the instruction INC B, will add one (increment) to the contents of the storage location B. While we can work this out from the abbreviated INC B, the microprocessor has to be told what to do in binary code.

The binary number 0100 1100 is the **machine code** instruction that tells the



1. Fetching an instruction from memory.



2. Addressing a given input device.

microprocessor to perform the function INC B. You must remember though, that not every processor will recognise the bit pattern 0100 1100 as the instruction INC B. Each type of microprocessor has its own instruction set and corresponding machine code specified by the manufacturer as part of its data sheet.

Different codes within the instruction set, naturally enough refer to different instructions. These codes are stored in computer memory in the order in which they are to be read (fetched), interpreted (decoded), and executed by the processor. A series of machine code instructions arranged in a specific order comprise the program.

The program steps held in memory must be located (addressed) by sending signals to memory to find whichever one comes next. The digital code that represents the location of each instruction in memory is known as its **address**.

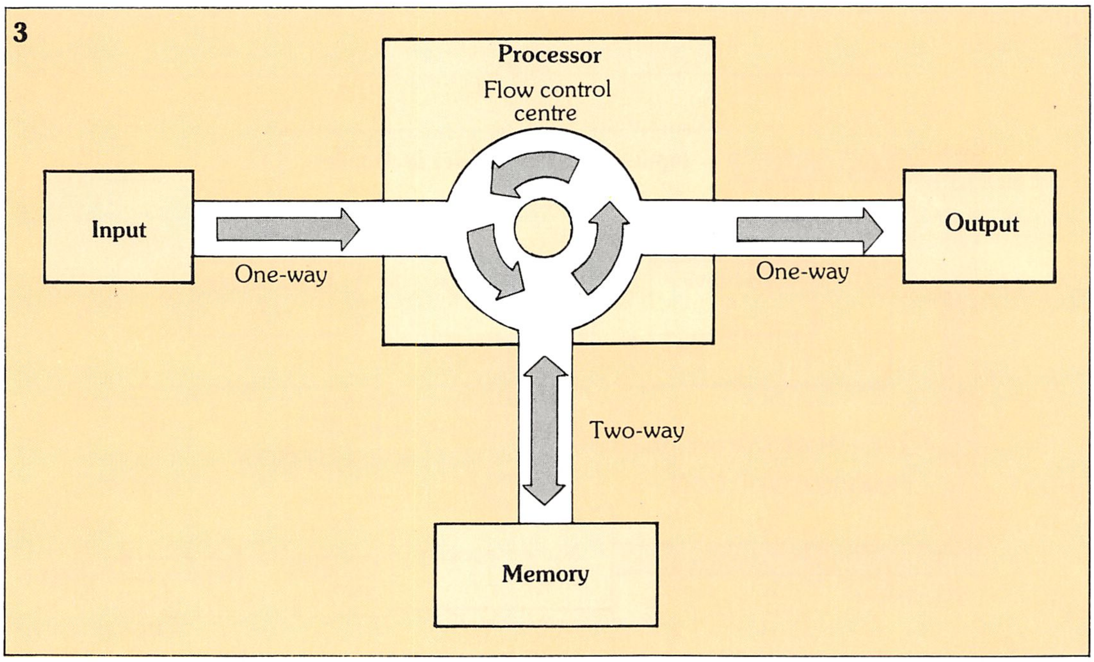
Address codes shouldn't be an entirely new concept to us. Remember, an address of a memory location is the code that represents where in the computer that memory location can be found. In a typical microprocessor, the address code can be 16 bits long, so up to 65,536 (2^{16}) different memory locations can be specified.

The procedure used to read or 'fetch' an instruction from memory is shown in *figure 1*. The microprocessor sends out the 16-bit binary code 0000 0011 1110 1000 that represents the decimal location number 1000 (which is the one that we want to access). At the same time, a digital control signal is sent to turn on the memory, so that it will read in the word of bits held at the memory location and transfer it to the processor. As *figure 1* shows, this signal is a **1** if the memory is to **read** the contents of the location, or a **0** if a number is to be **written** into memory.

The memory address decoder interprets the address code and sends back the stored information – in this case, the 16-bit instruction code 0000 0010 0000 0010. Once this has been received by the processor, the address code and memory are turned off.

This information, remember, is sent along **busses** or data communications channels which act like a number of wires in parallel. Wherever it is important, we shall indicate the number of bits that travel in a bus by a number to its side as on *figure 1*.

Figure 2 illustrates what happens when a given input device is addressed by the microprocessor. Suppose the processor receives the instruction 'input information from device 10' – probably abbreviated to something like, IN 10. The microprocessor decodes this and acts by sending the address code for 'input location 10' over the address bus.



3. Typical data paths within a microcomputer.

Simultaneously, it sends a 1 signal on the control line to turn on the input units.

As you can see from *figure 2*, there is an address interpreter inside the input subsystem which senses and decides which input device the information is required from. The input information – 1111 0000 1111 0000, in this case – is then transferred from input device 10 to the microprocessor.

Once received by the microprocessor, the address code and **input on** signals are removed by the processor, turning the input devices off. This completes the instruction, and you will have noticed the familiar pattern – sense, decide, act – at work again.

To output information, a similar sequence of events takes place, this time with an **output on** signal. Here the address code specifies which location in the computer is involved in transferring data to and from the microprocessor.

In many cases where input and output is involved, another memory control line is used. This is indicated by the dotted line in *figure 2*. When this line is at 0, the memory is on, and when it is 1, the memory is off. This allows the same address bus to be used for both memory and the input subsystem. When the control line is 1, the input is on and the memory is off and vice versa.

Figure 3 illustrates the flow of data along typical paths within a computer.

Timing signals and clocks

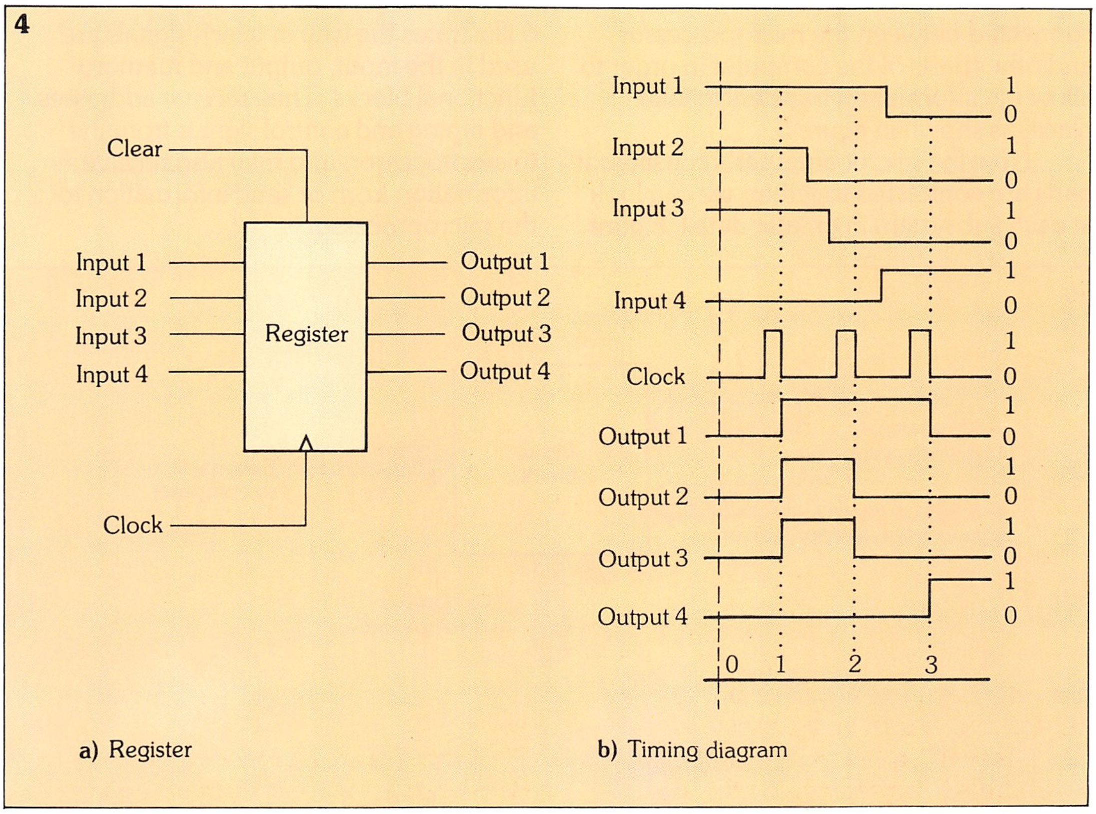
Before we go on to study these signals in more depth, we should look at the way in which they are synchronised.

The digital circuit shown in *figure 4a* is a 4-bit register which can store four 1 or 0 levels on its outputs to provide a temporary storage place. This circuit is built from flip-flops, which were considered earlier in *Digital Electronics 12, 13 and 14*.

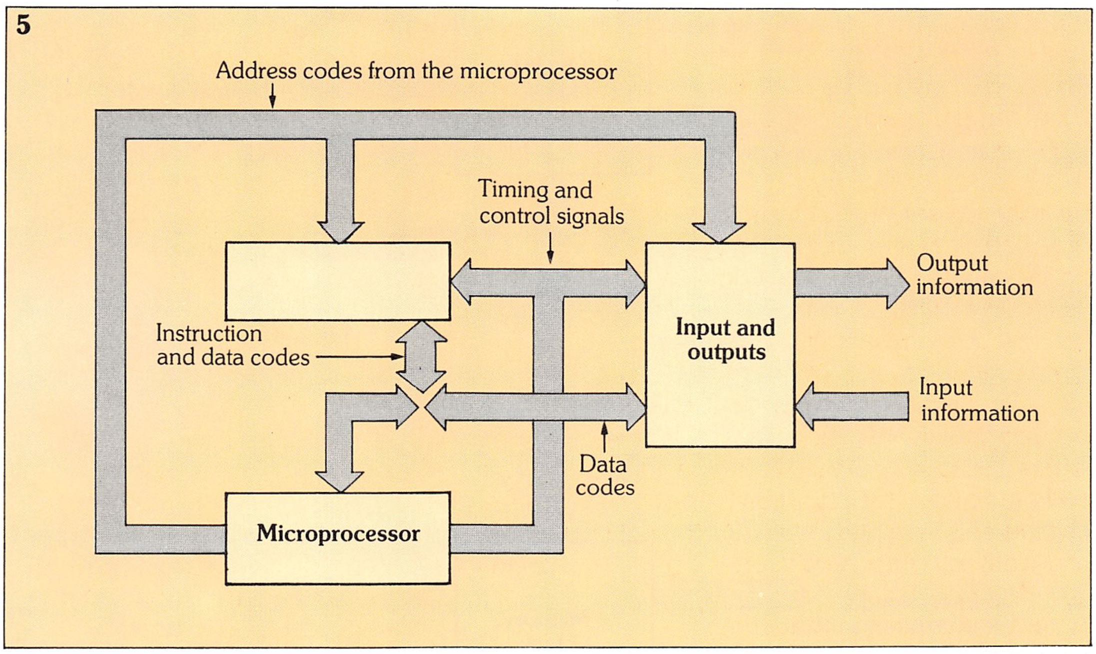
Flip-flops, as we know, will not change state (0 to 1 or 1 to 0) until a clock signal is applied. The timing diagram (*figure 4b*) shows the inputs and outputs of the register for three time periods. At time zero, signals 1, 1, 1 and 0 are applied to inputs 1, 2, 3 and 4 respectively. At the same time, all the outputs are at zero. At time one, triggered by the clock signal, the output code changes to 1, 1, 1, 0 – the same as the inputs. At time two, the input code is now 1, 0, 0, 0. The output code does not change to 1, 0, 0, 0 until it is triggered by the clock, even though the input lines changed at different times between time one and time two. Further changes in the input causes a code of 0, 0, 0, 1 at the outputs, triggered by the clock signal at time three.

Note that the clock signal occurs at regular intervals, which is where, of course, the name comes from.

4. System timing: (a) a 4-bit register built from flip-flops; **(b)** timing diagram.



5. Signal flow within a computer.



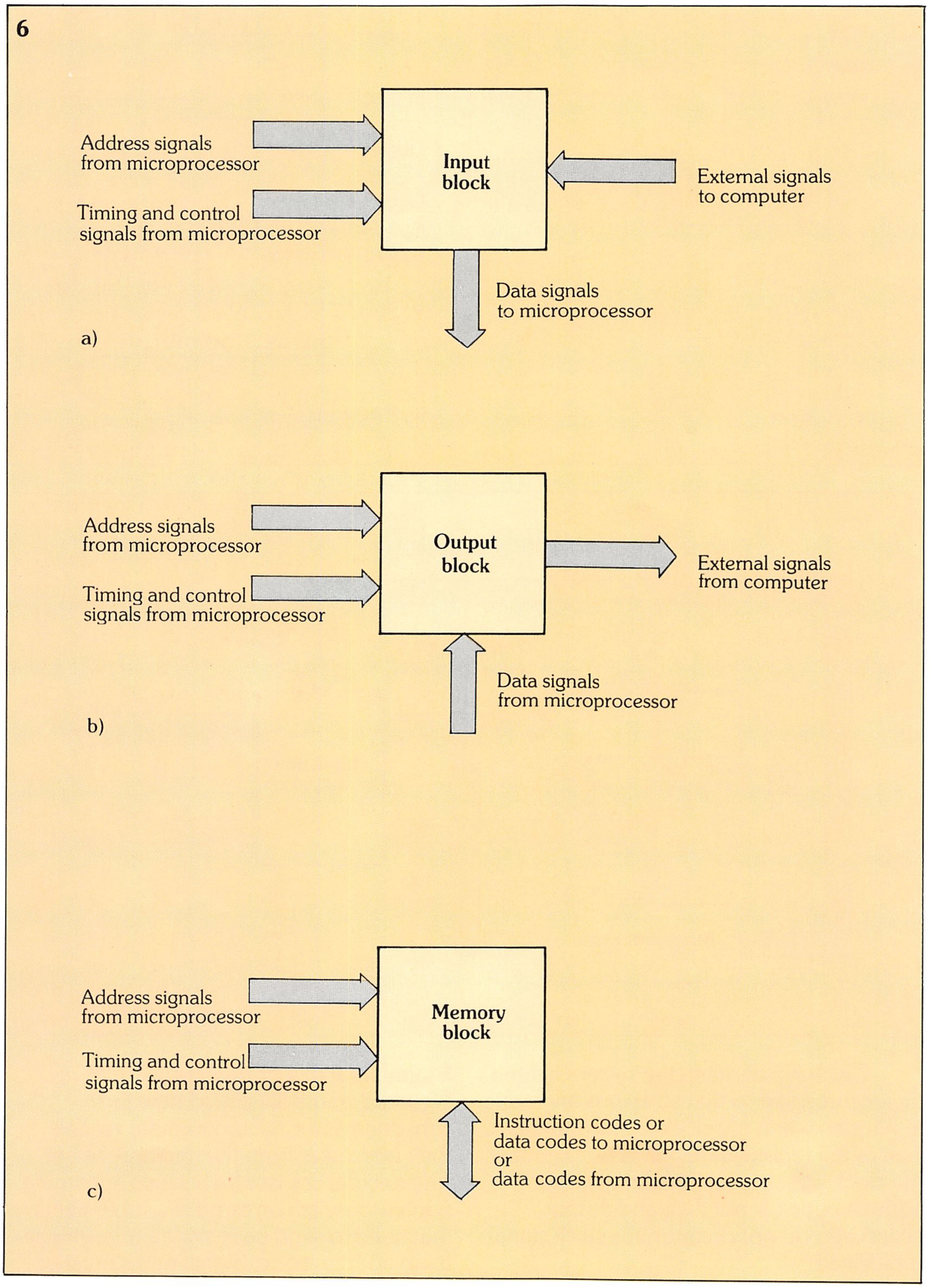
Clock signals keep the system in time or synchronised so that all events are made to happen when the clock signal appears. Just as the register changes its outputs at the clock signal, so control line signals, address codes, input data and output action also do not occur until a clock signal is received.

Signal flow in a computer

Figures 1 and *2* illustrated how the subsystems of a microcomputer receive their addresses from the microprocessor. The microprocessor then sends out timing signals to switch on or off the subsystems that need to send back signals in response to an address code. Signal paths must be

connected between the microprocessor and other parts of the computer in order to allow the information transfers to take place, as shown in *figure 5*.

Now the microcomputer's constituent parts are connected together, we can look at each subsystem in greater detail. *Figure 6* illustrates the way in which signals are used in the input, output and memory functional blocks. They receive addresses and timing and control signals from the microprocessor, and may also receive information from or send information to the microprocessor.



6. Signal flow at the: (a) input; **(b)** output; and **(c)** memory functional blocks.

Signal flow in the microprocessor

Having briefly examined the units that are connected around the microprocessor, let's look at the microprocessor itself. The addresses and timing and control signals sent to the computer's other functional blocks are generated by circuits in the microprocessor.

Address circuits

Depending upon whether the microprocessor is fetching an instruction from memory, or is transferring data to and from memory or other functional block, it must generate either an **instruction address** or a **data address**. The program counter, which is a storage register that sends out the memory address of the next instruction, is shown in *figure* 7. As we know, the program counter can be incremented so that successive instruction addresses can be generated.

Data is usually used when the microprocessor executes an instruction, and this must be located by an address either in memory or from an input. The data address may come into the processor as part of the instruction under execution, or it may have been 'saved' from an earlier instruction. It is held in the data address register.

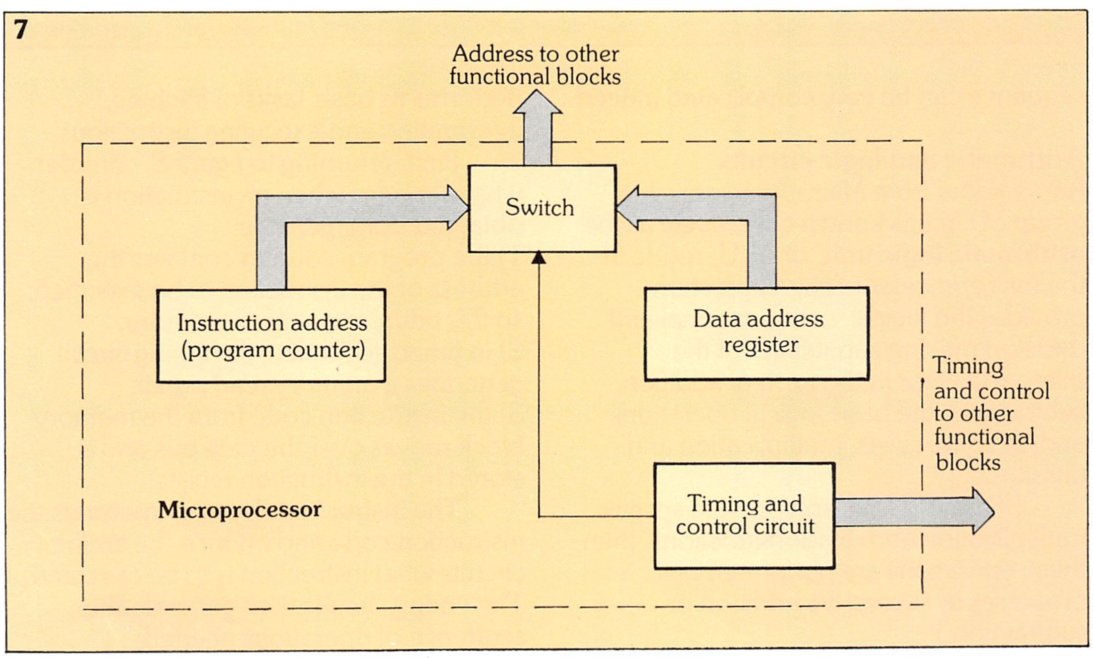
As both the program counter and the data address register can send signals down the address lines, the processor must be able to switch between them. This switch is controlled by the microprocessor's timing circuitry, so if the processor is about to fetch data from memory, the timing circuit switches the contents of the program counter onto the address bus. If the processor is about to send data to memory or an output device, then the timing circuit switches the data address registers with the appropriate address, onto the address bus.

Simultaneously, the timing circuit sends out the correct memory or input/output control signal, to switch on the subsystem involved in the transfer.

Each instruction executed by the microprocessor may involve a number of different steps, each of which may require data from one of the possible sources. The timing circuit thus has to switch the source of the address on the address bus from program counter to data address register in the correct sequence. The program counter is also automatically incremented after each instruction so that it contains the memory address of the next instruction to be executed.

Instruction decoder circuits

Once an instruction has been transferred to the microprocessor from memory, it is



7. Address circuits inside a microprocessor.

stored in the instruction register (*figure 8*) to be interpreted by the decoder.

The instruction decoder performs basically the same decoding operation for the instruction code as the address decoder for the address code in the memory and input/output circuits. However, as *figure 8* shows, there is a difference between the two. The output of the instruction decoder initiates signals from the timing circuits in a required step-by-step sequence to execute the instruction. Depending on the complexity of the instruction, this timing sequence can be very complicated indeed.

Arithmetic and logic circuits

As we know from *Microprocessors 1*, a group of circuits known collectively as the **arithmetic logic unit**, or ALU, reside in the microprocessor. This subsystem provides the logical, computational and decision making capabilities of the microprocessor in the form of addition, subtraction, the basic logical operations and, in some cases, multiplication and division.

(If the ALU doesn't provide specific multiplication and division functions, then these operations are performed by processes of successive addition or subtraction.)

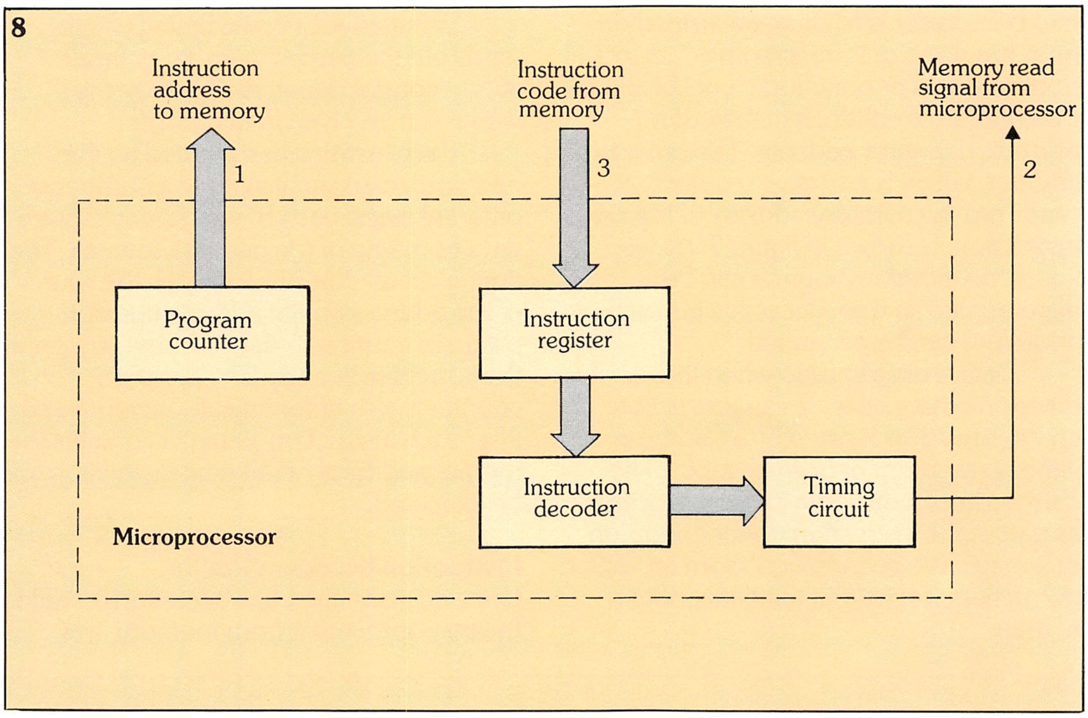
This subsystem will be examined in greater detail later in the chapter, but for now it is sufficient to know that the ALU contains registers providing temporary data storage, and that there is a flow of information between these registers and the computer's other subsystems as shown in *figure 9*.

Overall structure

Now that we have discussed the individual components of the microprocessor, let's piece them together to see how the device performs its basic tasks of fetching, interpreting and executing instructions.

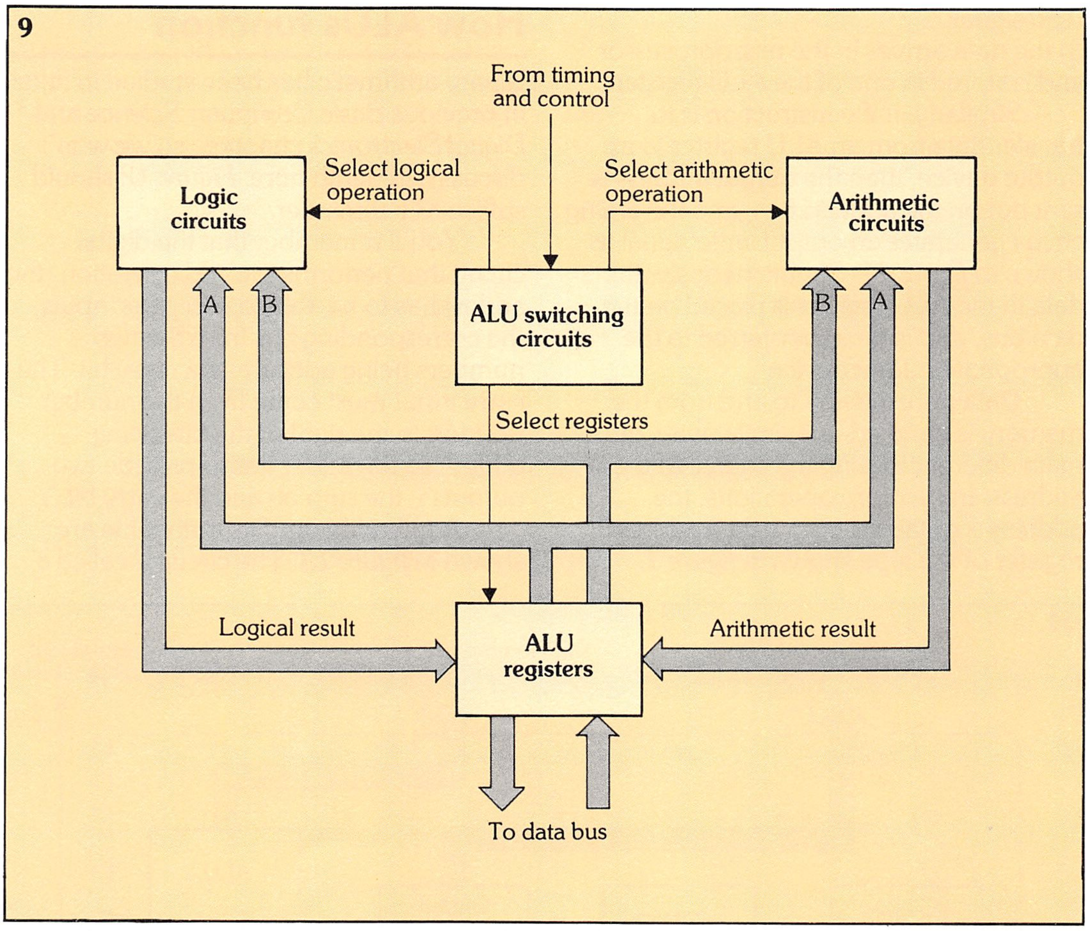
First, returning to *figure 8*, consider what happens when an instruction is obtained from memory:
1) the program counter contains the address of the instruction to be executed, so this address is sent to memory;
2) in timed sequence, the timing circuit generates a memory read signal;
3) the instruction code from the memory block moves over the data bus and is stored in the instruction register.

The instruction decoder interprets the instruction code and informs the timing circuits what instruction is to be executed. The timing circuits then generate the sequence of operations needed.

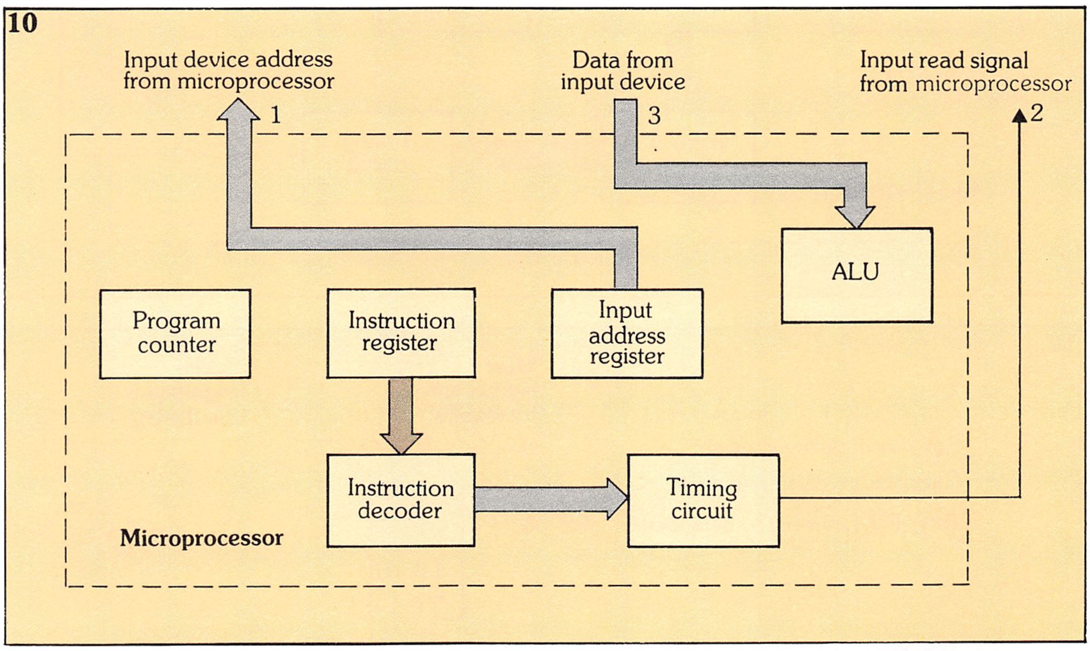


8. Unlike the address decoder, the instruction decoder initiates signals from the timing circuits to execute an instruction.

9. The ALU subsystem within a microprocessor. There is a flow of information between the ALU registers and other subsystems via the data bus.



10. Operation of an instruction to transfer data from an input device to the microprocessor.



If the instruction involves the transfer of data from an input device to the microprocessor, then the events shown in *figure 10* would occur in this sequence:

1) the address of the input device is obtained from the instruction code and is sent out on the address bus;
2) the timing circuits generate an input

read signal;
3) the data arrives in the microprocessor and is stored in one of the ALU registers.

Similarly, if the instruction is to transfer data from an ALU register to an output device, then the output address is sent out on the address bus, and the timing circuit generates an output write signal as shown in *figure 11*. Simultaneously, the data in the ALU register is placed on the data bus, and is thus transferred to the appropriate output device.

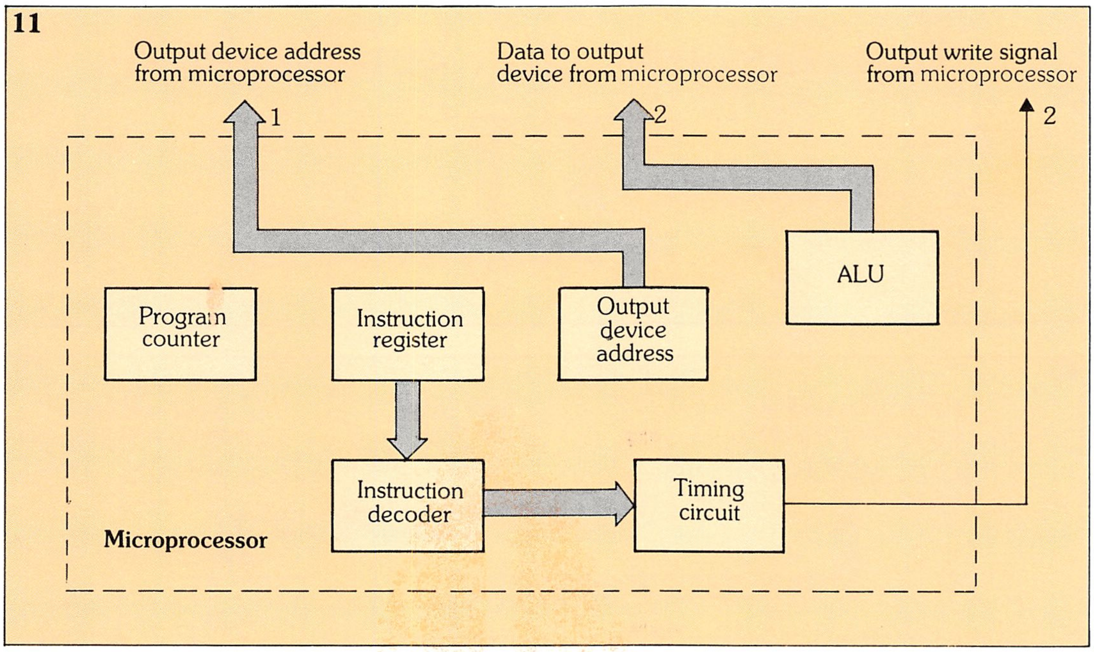
Data transmission to and from the memory is handled in a similar way, the main difference being the source of the address. In *memory* operations, the address is obtained from a data address register of the type shown in *figure 7*.

How ALUs function

Binary arithmetic has been studied in detail in previous *Basic Computer Science* and *Digital Electronics* chapters, so we won't discuss it in depth here. *Figure 12* should suffice as a reminder.

You'll remember that the digital circuit that performs the add operation (the adder) has to be able to accept as inputs the corresponding bits from the two numbers being added and a carry bit. The carry input must come from the number position to the right of the bits being added, so the adder has to provide two outputs – the sum bit and the carry bit.

A full-adder and its truth table are shown in *figure 13*. The circuit is called a



11. **An instruction to transfer data** from an ALU register to an output device is executed in this way.

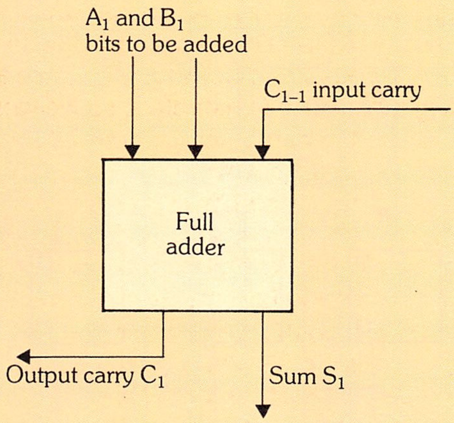
12

	Most significant bit (MSB)			**Least significant bit (LSB)**	
Equivalent decimal weight	8	4	2	1	
Carries from bit sum to right	1	1	0	0	
Bits of first number to be added	0	1	1	1	(4 + 2 + 1 = 7)
		10			
Bits of second number to be added	0	+1	1	0	(4 + 2 = 6)
Sum bits	1	1	10	1	(8 + 4 + 1 = 13)

12. An example of binary addition.

13. A full-adder and its truth table.

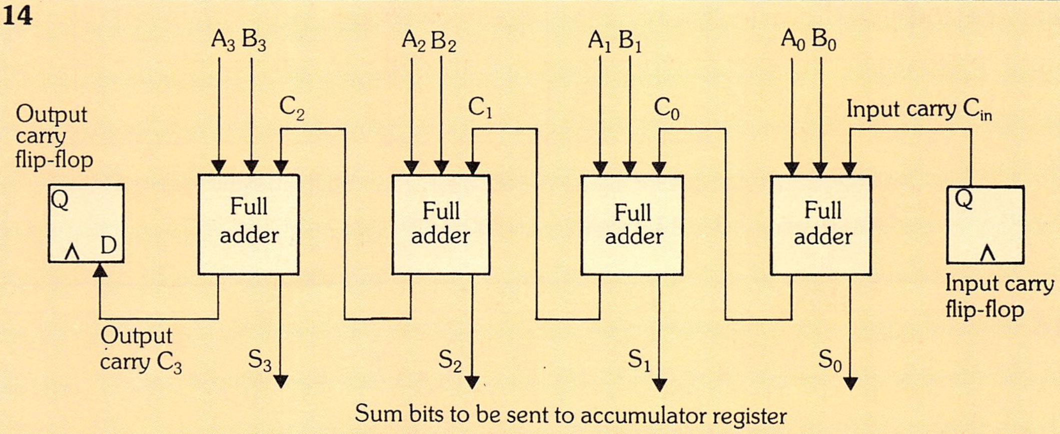
13



Inputs			Outputs	
A_1	B_1	C_{1-1}	S_1	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

14. A 4-bit adder circuit.

14



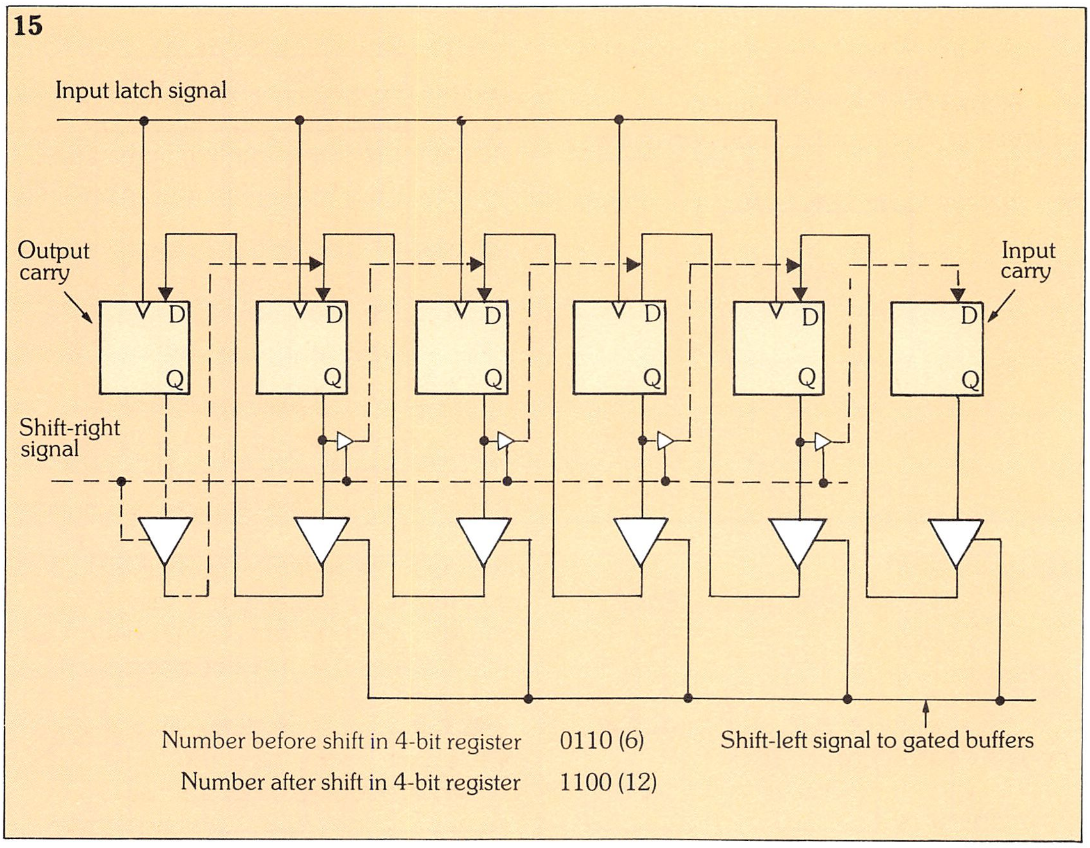
full-adder because it can accept a carry bit as input. Single bit adders such as this can be cascaded into 4, 8 and 16-bit arrangements, and this principle is used in the microprocessor's ALU.

A 4-bit arrangement is shown in *figure 14*. Provision is made for a flip-flop to hold the input carry to the least significant bit (LSB) S_0; this carry may result from some previous addition. A flip-flop is also provided to hold the carry produced by the addition of the two most significant bits (MSB). However, if the adder timing and control circuitry is designed properly, both of these flip-flops can be combined. The control circuitry also determines how many different registers can be used to send numbers to the adder, and where the sum is to be sent.

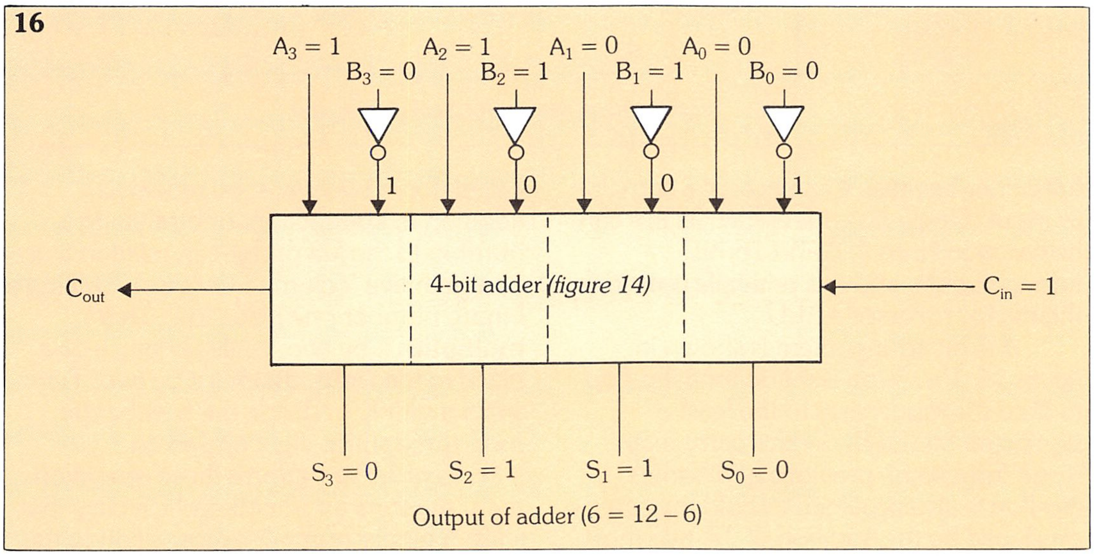
Multiplication and division

We now know that multiplication and division can be achieved by a computer or calculator in two ways: by repeated addition or subtraction; or by shifting a number to the left or right, in relation to its denominitive 'column headings'. Shifting a binary number one place to the **left**, **multiplies** it by two; while shifting it one place to the **right**, **divides** it by two. Two shifts multiply or divide the number by four, three shifts, eight and so on.

In order to perform these operations then, registers and/or control circuitry that shifts a binary number one position to the left or right is needed. *Figure 15* illustrates just such a register. To perform a shift left, a 1 is placed on the SHIFT-LEFT signal line and the output of each flip-flop is sent to the input of one to the left by the gated buffers when the clock signal is received. The input carry flip-flop's output is sent to the first flip-flop, to set the LSB bit, while the output carry flip-flop holds the MSB that was present before the shift, as it may be needed for a subsequent operation.



15. Multiplication and division in a shift register.



16. Subtraction using binary adders.

A shift right is performed by gating the buffers with a 1 on the SHIFT-RIGHT signal line, to send the output of each flip-flop to the one on the right.

If the number stored in the register is 0110 (binary equivalent of decimal 6) before the left shift, the number stored after the shift operation is 1100 (binary equivalent of decimal 12). The shift left has multiplied the number by two. If the 1100 is now shifted right to become 0110, the 12 would be divided by 2, giving us 6.

Subtraction

In order to limit the amount of circuitry needed by the microprocessor, the adder in the ALU also performs subtraction. This is achieved by changing the bits of the subtrahend (number to be subtracted) to their inverse values, adding 1 to the result

(thus giving the two's complement of the number), and then adding the result to the minuend (number to be subtracted from).

Figure 16 illustrates how this is achieved using the inverter on each of the subtrahend's adder input lines. By inverting the B bits before they are input to the adder, and by providing an input carry of 1, the adder subtracts. As you can see, the example shown in *figure 16* subtracts 6 (0110) from 12 (1100):

$$\begin{array}{rl} 1100 & (12) \\ \underline{-0110} & (6) \\ \underline{0110} & (6) \end{array}$$

which is carried out as follows. The two's complement of 6 is 1010 and:

$$\begin{array}{rl} 1100 & (12) \\ \underline{+1010} & \text{(Two's complement of 6)} \\ \underline{0110} & \\ 1 & \text{(carry)} \end{array}$$

which gives us the correct answer, as we ignore the output carry.

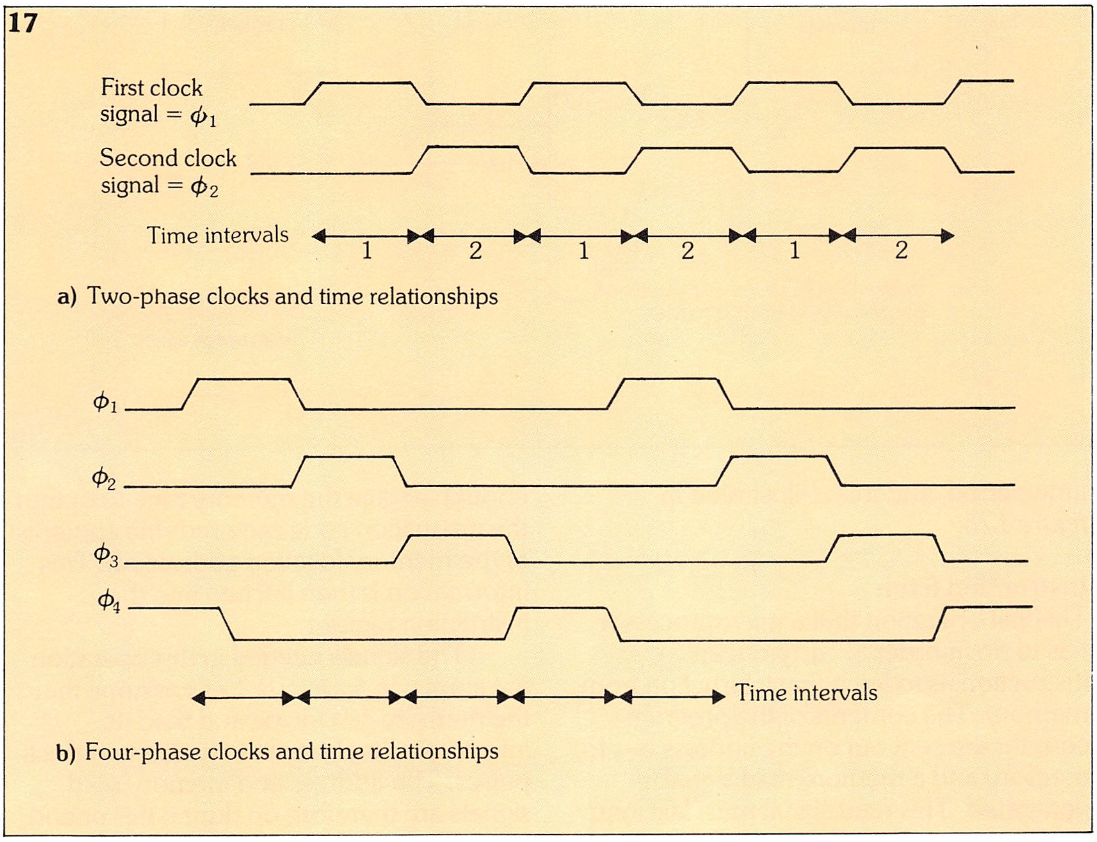
As you can see, these arithmetic operations are simple, but repetitive, so fast operation times are essential to computers.

Timing and control functions

The microprocessor's timing and control centre has to turn all the circuits within the chip on and off as well as controlling the memory and input/output functional blocks. Because the timing circuits provide the control for the whole system, they are one of the most important parts of the microprocessor.

Every operation that we have discussed so far needs a series of events to occur in a synchronised step-by-step manner. To read memory successfully, addresses must be sent to memory, a signal must be sent to perform the read, and then, after enough time delay to ensure that the information has been read, the bits must be sent to the proper destination.

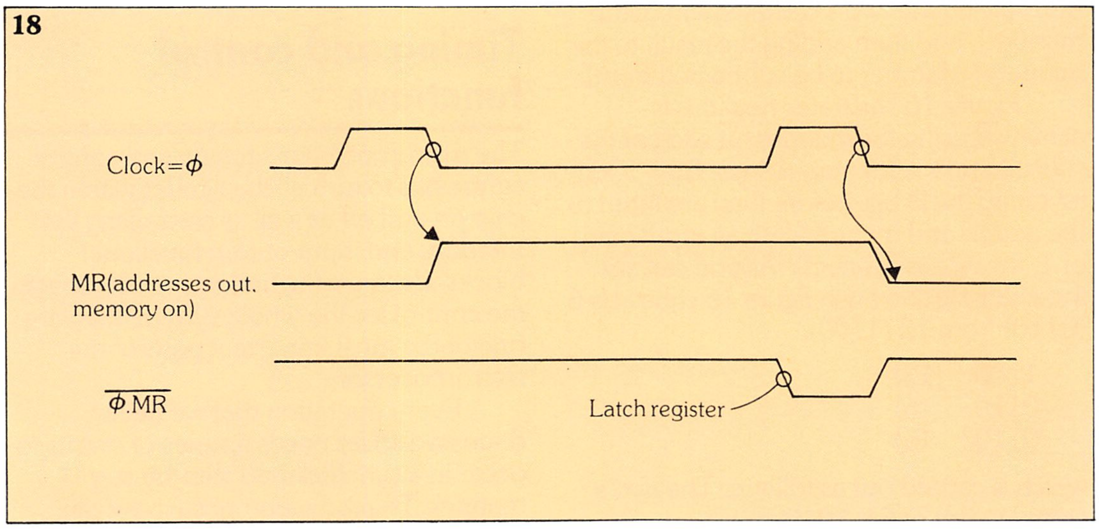
One or more clock signals are used for this master timing – some microprocessors use a two-phase clock (*figure 17a*) and divide time into smaller intervals as needed. Other microprocessors use four phases to provide the required



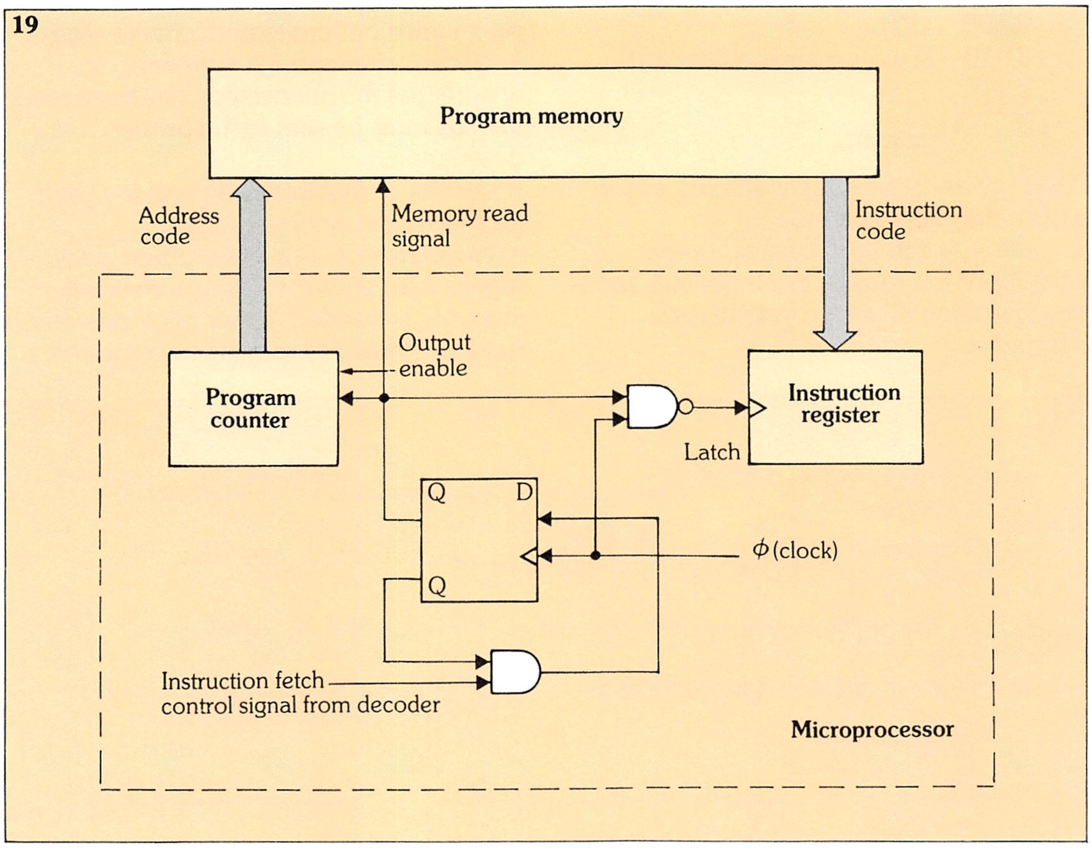
a) Two-phase clocks and time relationships

b) Four-phase clocks and time relationships

17. Master clock signals: (**a**) two-phase; (**b**) four-phase.



18. Timing signals required for a memory read operation.



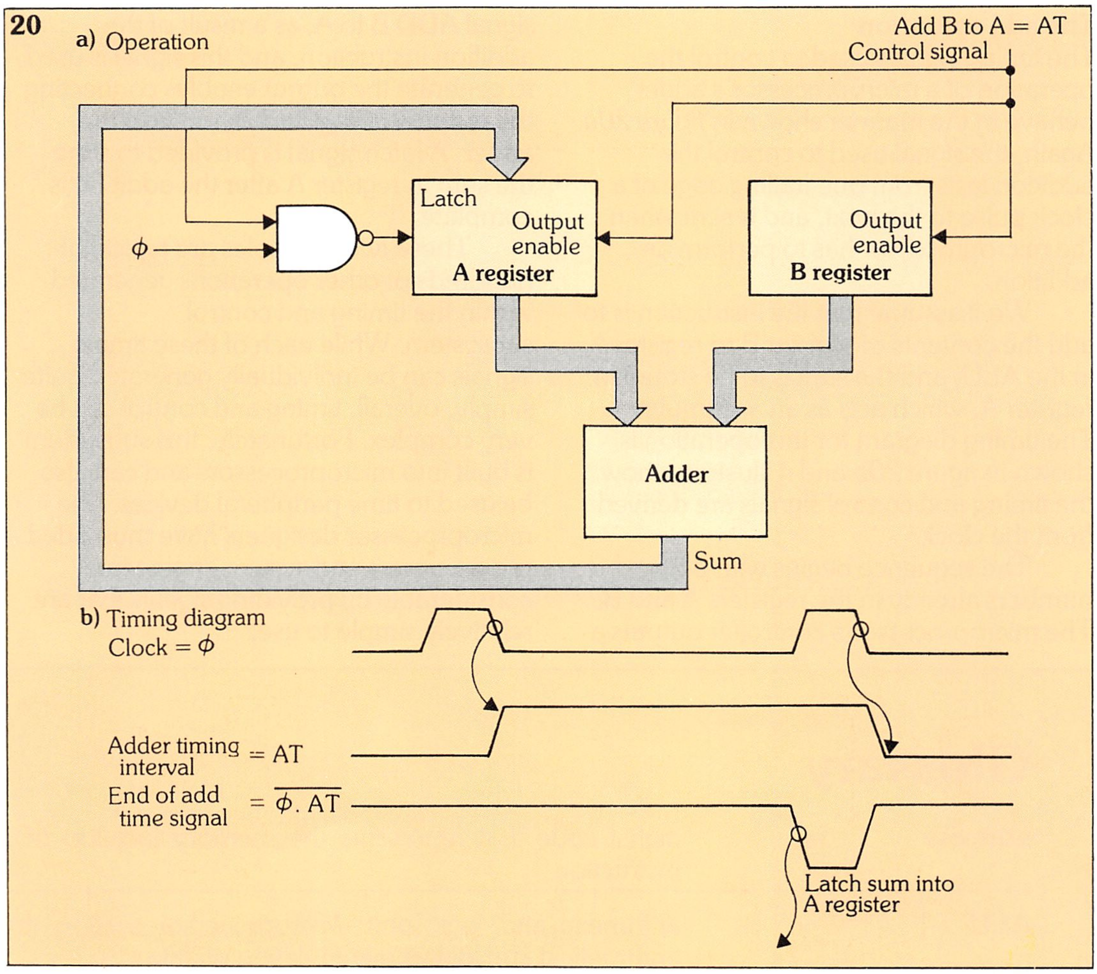
19. Timing and control circuit for a memory fetch operation.

time period, and this is illustrated in *figure 17b*.

Instruction fetch

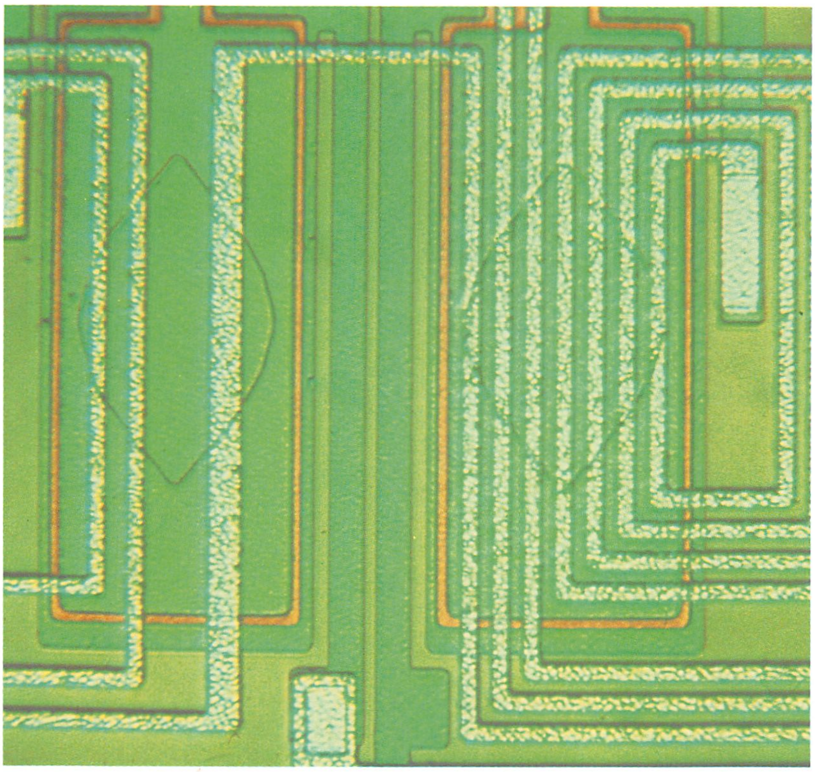
The first operation that a microprocessor has to do in order to carry out an instruction, is to fetch that instruction from memory. The contents of the program counter are sent out on the address bus to memory and a memory read signal is generated. This read signal must last long enough to give the memory time to output the instruction code required (the contents of the memory location addressed). This information is then latched into the instruction register.

The signals needed in this operation are shown in *figure 18*. Let's assume that the memory can locate and read its information in the time between two clock pulses. The address and memory read signals are therefore on during this period.



20. (a) Operation of the timing signals used to control the microprocessor's adder; **(b)** timing diagram showing how the timing and control signals are derived from the clock.

Below: electron micrograph of a Josephson junction. Considerable research has gone into the development of a production technique for these high speed devices to produce super-fast computers. As yet, no solution has been found.



A signal must latch the information into the instruction register at the end of this time period, but before these two signals have returned to zero.

Figure 19 shows how the circuits derive these signals from the clock signal. The memory read signal is formed using a clocked D flip-flop that stores a 1 on the trailing edge of the clock, and that will trigger again on the next clock trailing edge, giving a 0 output. This control signal is sent to the program counter's output enable, so that the instruction address is sent out on the address bus.

The same control signal is also sent to memory, as a memory enable or memory read signal. MEMORY READ (MR) is connected to a clock via a NAND gate to cause the instruction register to latch information from memory at the proper time. So we can see that the clock signal – along with information from the microprocessor controller – is used to provide the necessary control and timing signals.

The add operation

The timing signals used to control the operation of a microprocessor's adder behave in the manner shown in *figure 20a*. Again, the signal used to control the addition lasts from one trailing edge of a clock pulse to the next, and is sent when the microprocessor has to perform an addition.

We'll assume that the instruction is to add the contents of register B to register A, in the ALU, and the sum is to be stored in register A, which acts as an accumulator. The timing diagram for this operation is shown in *figure 20b* and it illustrates how the timing and control signals are derived from the clock.

The sequence begins with both numbers already in the registers A and B. The microprocessor's controller outputs a signal ADD B to A, as a result of the addition instruction, and this signal is used to generate the output enables connecting the outputs of registers A and B to the adder. A latch signal is provided to store the sum in register A after the addition is complete.

These two examples are typical of hundreds of other operations generated within the timing and control subsystem. While each of these timing signals can be individually generated quite simply, overall, timing and control can be very complex. Fortunately, this subsystem is built into microprocessors and can also be used to time peripheral devices. The microprocessor designers have thus aided programmers and system engineers considerably by providing signals that are relatively simple to use.

Glossary

address	digital code that represents the memory location of a data item or instruction
ALU	arithmetic and logic unit. Microprocessor section that controls the arithmetical and logical manipulation of data
clock	device that sends a regularly oscillating signal to the microprocessor's timing and control centre, to provide the synchronisation and control essential for correct operation of the complete system
instruction code	pattern of bits that control a microprocessor's operation
latch	binary register or store
machine code	the language that a computer can understand and use
shift register	a register in which the stored data can be shifted to the left or right. Used for multiplication or division by powers of two

10 COMPUTERS & SOCIETY

Automated manufacturing

Structure of manufacturing industries

Product as diverse as carpets and cars, furniture and food are, of course, all manufactured from different raw materials which involve different industrial processes. However, the pattern behind these processes and the computers which are used to control them are often very similar. To recognise these similarities we shall have to take a step back and find the common features of manufacturing industries.

All manufacturing industries process raw material or components, turning them into products to sell at a profit to their customers. To ensure that customer demands are met, marketing and sales departments exist to keep companies in touch with their market-place. The knowledge acquired from this research is then used to identify new products or improve existing products – in the hope that consumers will be satisfied.

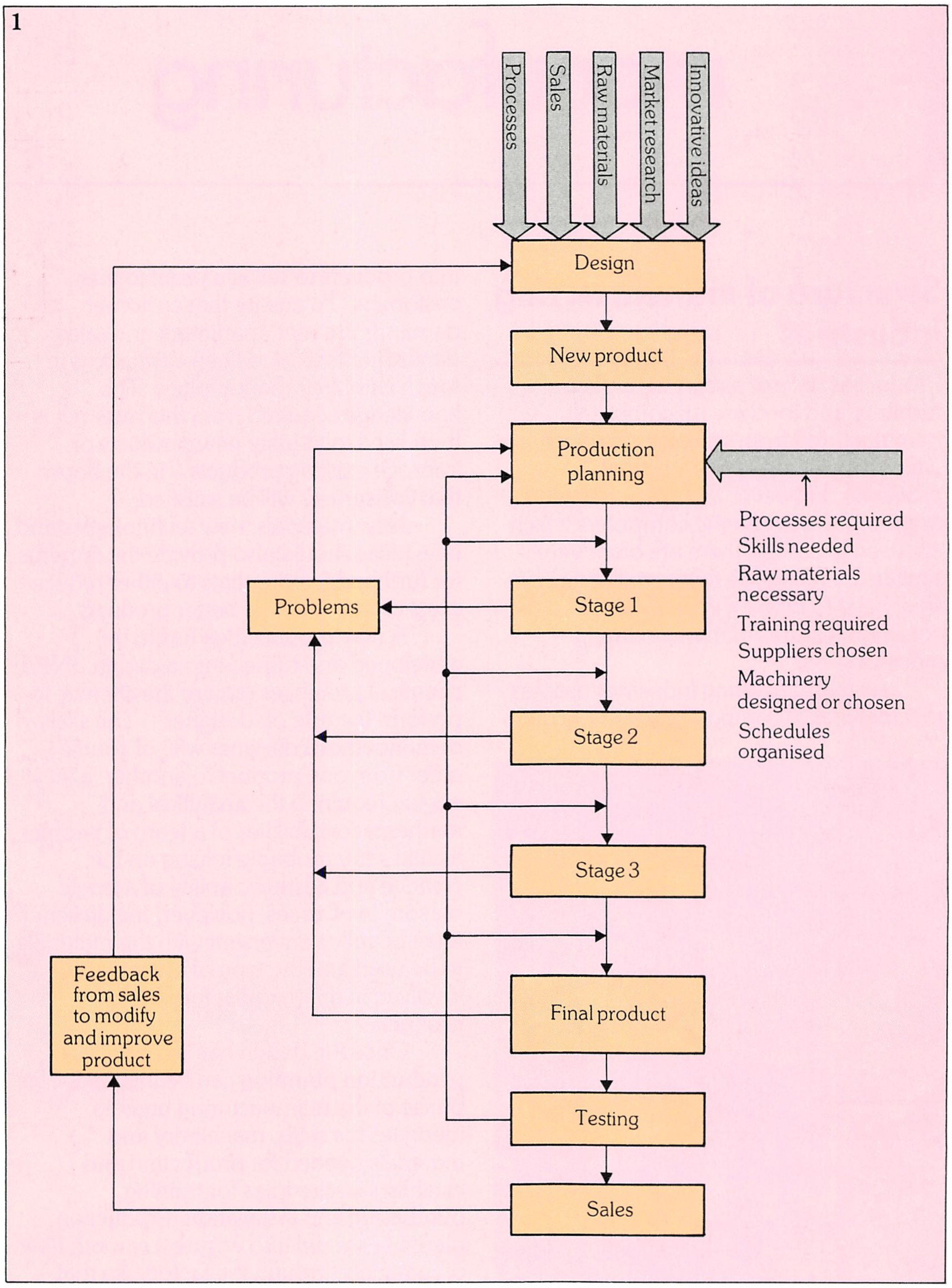
Below: this Workmaster R6-140 robot has a six-axis articulated arm that can handle 140 kg, moving through a horizontal arc of 240°. It can be programmed with up to 1500 lines plus 500 positional co-ordinates. (Photo: Thorn EMI/ Muirhead).

New materials, new technologies and new ideas should also provide the impetus for further developments to either reduce costs or manufacture better products.

A new product idea has to be developed and turned into a design. (New chemical processes require the chemist to perform the role of 'designer'.) The skills demanded of a designer will, of course differ from one product to another: a jet engine requiring the analytical and mathematical abilities of a team of people; a child's toy probably relying on the creative and aesthetic ability of a single person. In all cases, however, the designer must be fully conversant with the materials to be used and the type of machinery involved in the manufacture of the product.

Once the design has been finalised, **production planning** can be initiated. This phase of the manufacturing process identifies the skills, machinery and materials needed for production and establishes schedules for training, purchasing and installation. Production planning should also ensure a smooth flow of materials through the factory, so that bottlenecks and delays in manufacture are avoided.

The need for new equipment or tools that have to be designed and built must also be recognised at this stage. Different material and component supplies are assessed and chosen, while contingency plans are made to cope with delays in supply and machinery breakdown.



1. Stages in the manufacture of a new product.

Carefully kept records of suppliers and their history of quality and reliability, together with maintenance records for machinery prove invaluable to successful production planning.

As the new product is finally rolling off the production line, close monitoring of the manufacturing operation reveals any flaws in the process(es) – necessary adaptations are implemented and efforts are made to optimise the processes involved.

Production control then, concerns monitoring production, checking plans and making adjustments if things seem to be going wrong.

The various stages in the manufacture of a new product are summarised in *figure 1*.

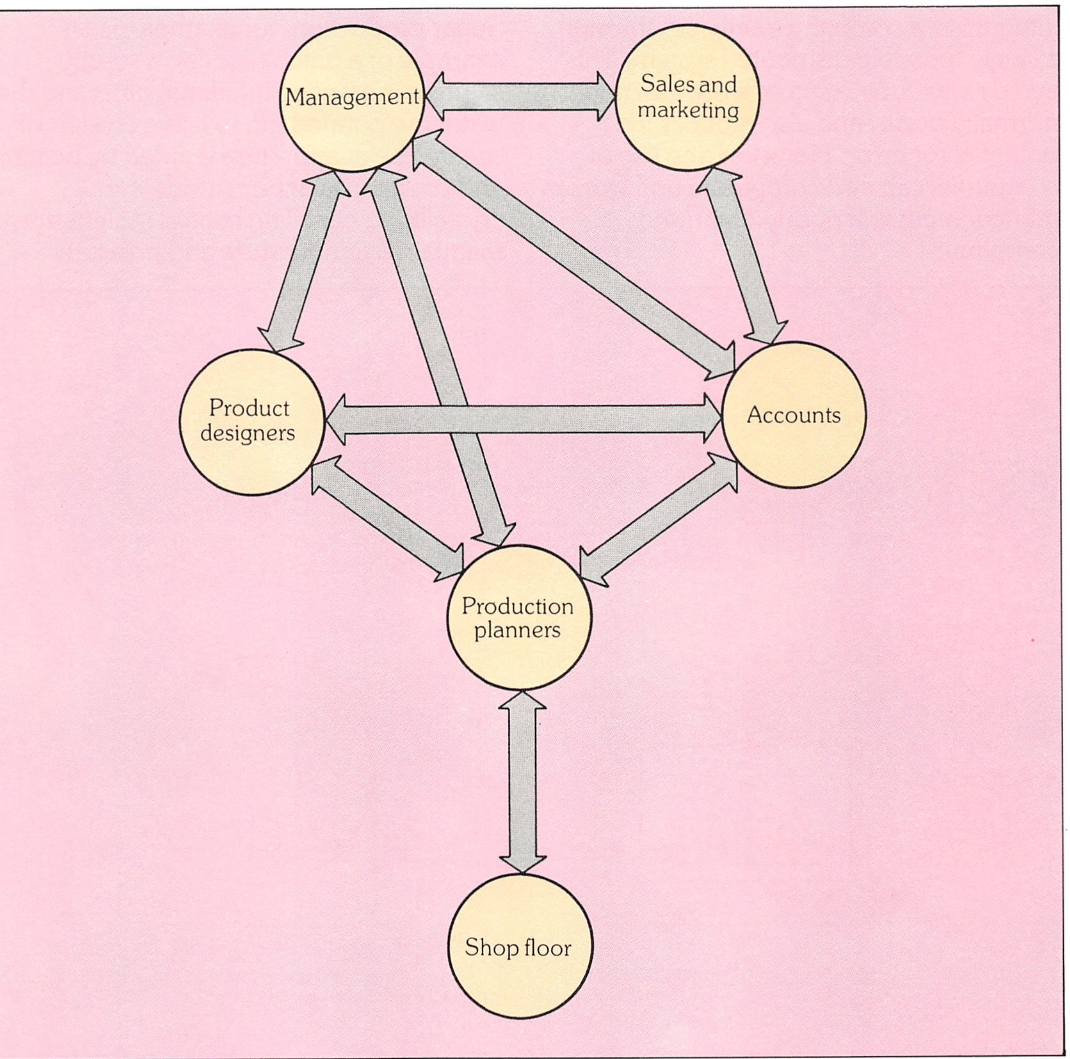
Types of production
Products usually undergo several stages of manufacture. For example, to make biscuits, dough has to be mixed and rolled thin; biscuits are then cut, baked, cooled, weighed and packed before they are transported to a warehouse or distribution point. This type of production is known as **continuous production** because the product is continuously transformed as it moves from one specialised machine or worker to another.

Electronic circuit assembly, on the other hand, involves inserting a variety of components in their respective places on a printed circuit board and soldering the connections – this is an example of **batch production**. Batches of sub-units of the whole product are produced and stockpiled, to be assembled into the final product at a later stage. This usually takes place at one location.

There are five basic activities that take place on the shop floor during the manufacture of a product:
1) **Processing**, where the materials are tranformed by shaping, coating, joining or chemically reacting.
2) **Assembly**, where a number of components are put together.
3) **Transportation**, where components are moved for further processing or storage.
4) **Inspection and testing**, where the quality and operation of components or products are checked.
5) **Waiting**, where components are held for further processing or where products are warehoused or displayed.

2. Information flow within a manufacturing company.

2



In order to support this manufacturing operation company administrators will: order raw materials; chase late deliveries; keep accounts; pay wages and salaries; provide company reports and manage personnel. Sales staff also keep records, inform customers of new products and provide feedback to the company that current products are those that are in demand.

From this brief outline it is obvious that information is being handled at every stage of production and operation. This information flow within a company is illustrated in *figure 2*. The sheer bulk of all this data usually necessitates the use of a computer or computers in order that it can be properly managed. It is vitally important that the right information reaches the right people at the right time and that information can be quickly interpreted.

Computers offer a means of integrating a company's activities through a single, possibly distributed system. This avoids any duplication of work that might normally occur, and also reduces the potential for error. However, the use of computers can also complicate and worsen management in a poorly organised company.

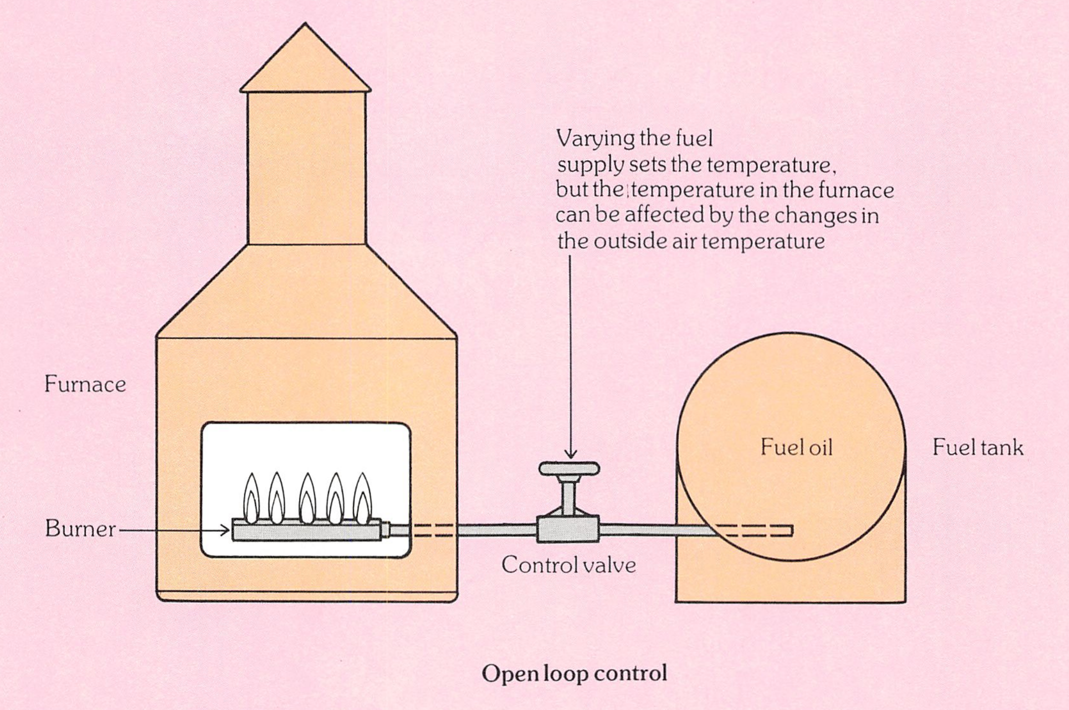
Process control

Process control is a term commonly applied to the automatic control used in chemical industries. It is usually applied to processes that transform materials through heating, cooling and chemical reaction. It deals with materials that can flow, and many of the methods of control deal with controlling flow.

The refinement of oil, for instance, involves heating, distillation and mixing with steam – reactions stimulated by catalysts and controlled cooling. The aim of process control is to ensure steady, well defined pressures, temperatures and oil flow in spite of any variations that might occur in the activity of the catalyst and the temperature of the cooling water or heating steam.

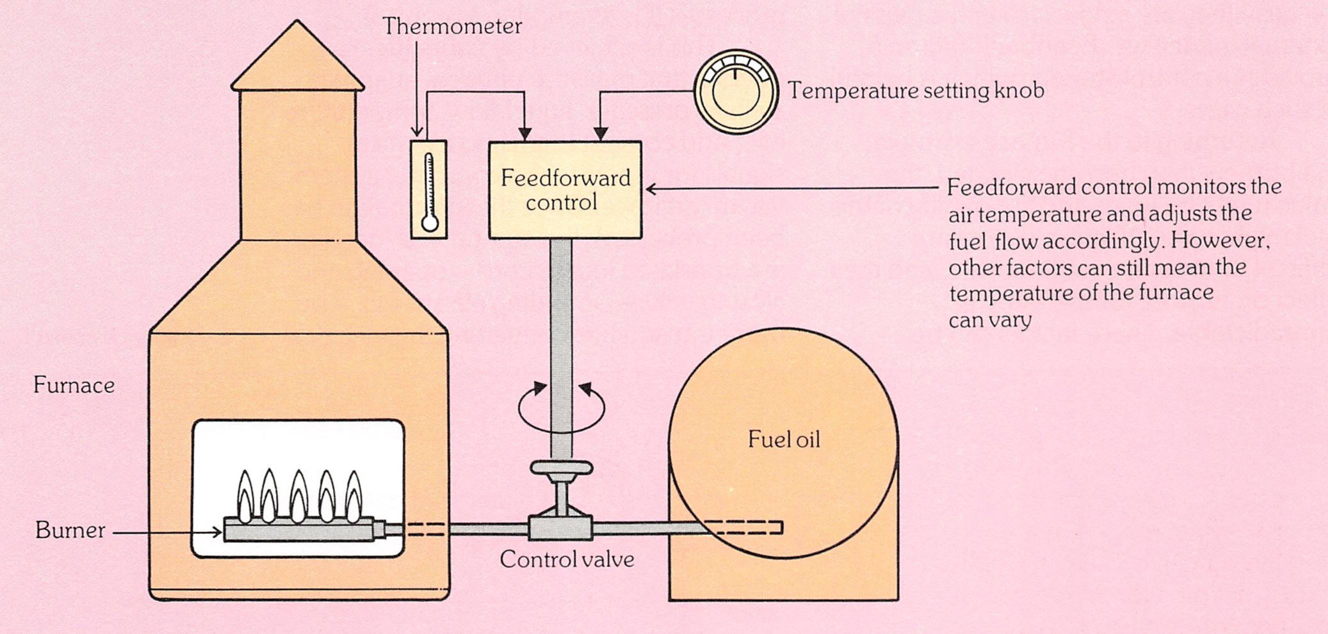
Both batch and continuous production exist in process industries. Sugar production, for example, is an example of a batch process: a syrupy solution is emptied into large pans and the water evaporated off, to leave crystals of sugar. The evaporation is aided by heating and creating a vacuum inside the crystallizing pan. The control system must maintain a temperature and pressure

3



3. Open-loop control.

4



Feedforward control

4. Feedforward control.

within very fine limits as these affect crystal growth.

Glass manufacture, on the other hand, is a continuous process. A mixture of materials, minerals and (sometimes) recycled glass, are fed continuously into a furnace. Molten glass overflows from the furnace onto the surface of a pool of mercury where it begins to solidify. The hardening ribbon of glass is then drawn out to the correct thickness on rollers, and cooled further, before being cut into sheets and transported to the warehouse. Here, the control system has to monitor the level of molten glass in the furnace, the speed of the rollers and the fuel and air being fed to the furnace's burners. Steady operating conditions must be maintained so that the glass is of consistent quality and thickness.

Computer systems can also be used to cut the glass to the sizes required by customers, and to cut these sheets around any blemishes (detected by photo-electric devices) in the glass surface.

Three types of control

An air fired furnace can provide us with examples of the three main types of process control. The flow of fuel oil to the furnace's burners is regulated by a valve. Setting the valve to a particular position provides temperature control; changing the setting alters the flow of oil and the temperature. The valve settings could be calibrated so that an operator is able to choose a temperature by adjusting the valve. This is known as **open-loop control** (*figure 3*).

However, this is not quite sifficient – if the air temperature outside the furnace drops, then the temperature inside the furnace also drops. Some mechanism is needed to increase or decrease the fuel flow accordingly to any variations in outside temperature. **Feedforward control** (*figure 4*) provides this mechanism.

Feedforward control involves measuring any external disturbances that affect the quantity being controlled, and applying compensation on the basis of the values received. In our example, the temperature of the air around the furnace is monitored and the valve setting (which modifies fuel flow and hence furnace temperature) altered accordingly, thus maintaining a steady temperature in the furnace. Analogue-to-digital conversion techniques are used so that a computer can perform these calculations.

Some processes cannot be controlled

accurately, even by feedforward means, because they are subject to such a mass of external influences. **Feedback** (*figure 5*) provides a control strategy which is helpful in such cases.

Returning to the furnace example, the burners may become clogged, the valve might be worn and the quality of the fuel might vary. All these factors are difficult to measure continuously, and their effect on furnace temperature is unpredictable. These factors can be contended with if the temperature inside the furnace is measured and compared with the temperature desired. If these two values are different, then the magnitude of the error is used to calculate the required compensation in the valve setting. This, of course, can be performed by a computer controller as a continuous process of sampling and adjustment.

Controllers

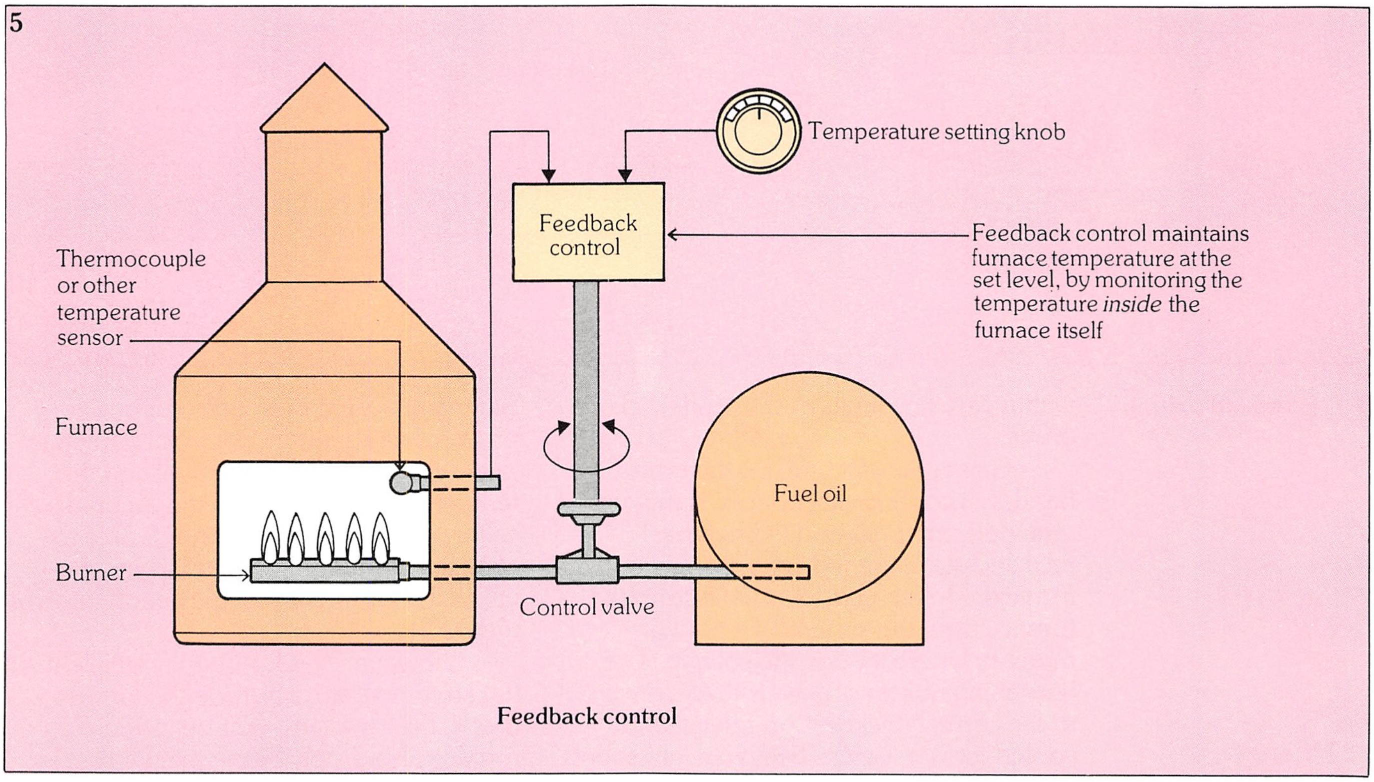
The controller (the heart of any process control system) is only capable of taking measurements, performing calculations, storing information and deciding on a course of action. However, for it to be useful in any automated control system, the controller must be in contact with the processes it is attempting to control.

This is achieved by **transducers** – devices that measure physical quantites, such as pressure, liquid flow, temperature etc., and convert them into electrical signals for inputs to the computer system via an interface. Once the information has been processed, the computer's 'decisions' are translated into actions – opening and closing valves, sounding alarms etc. – by more transducers converting the electrical signals from the computer into action.

As we saw in *Digital Electronics 9*, most 'real world' mechanical or electrical systems work in a linear or analogue manner, while computers are digital systems. The transducers in a process control system deal with analogue signals, so the interface between controller and transducer takes the form of analogue-to-digital and digitial-to-analogue conversion (ADC and DAC). The process is summarised in *figure 6*. (ADCs, DACs, and transducers are covered in detail in *Digital Electronics 19* and *22* respectively.)

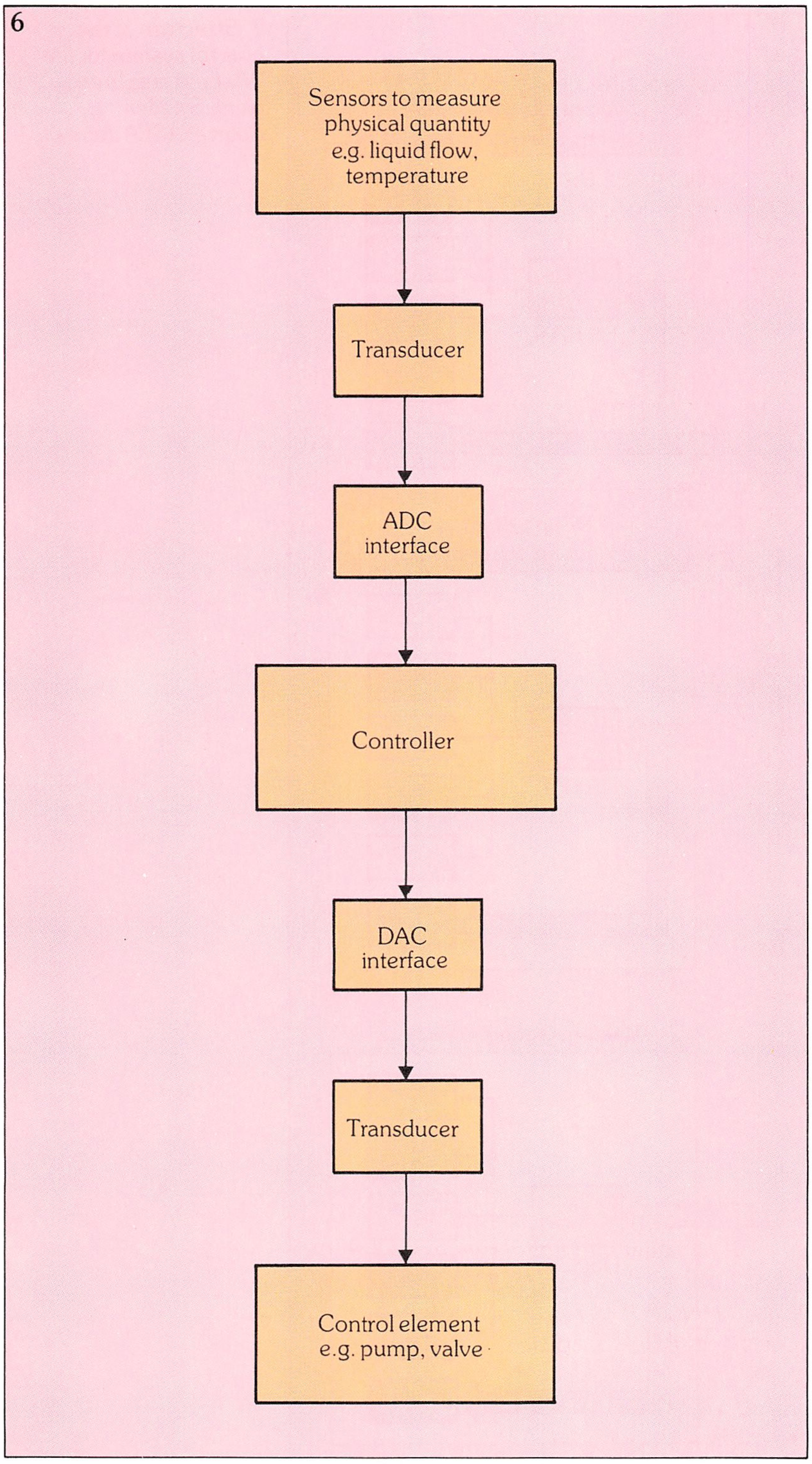
While in some processes only one or two quantities need to be controlled, in others there can be hundreds of quantities.

5. Feedback control.



Feedback control

6



6. How the controller keeps in contact with the processes it is controlling.

Some controlled quantities are also dependent on others, for example in the furnace control system, air flow to the burners is related to fuel flow thus ensuring efficient pollution free combustion.

The problem of effective communication between different controlled processes is solved in two ways. In sluggish, slowly acting processes, control calculations are performed infrequently and so one relatively fast computer is able to share its time amongst many processes. For faster processes, one computer may not have sufficient time to perform all calculations and so several communicating computers are used.

Multiple computer systems have the advantage in that the system is then guarded against failure by having back-up capacity to hand. The disadvantage, however, is that multiple computer set-ups demand that more time is spent on inter-computer communication than on calculations and hence control.

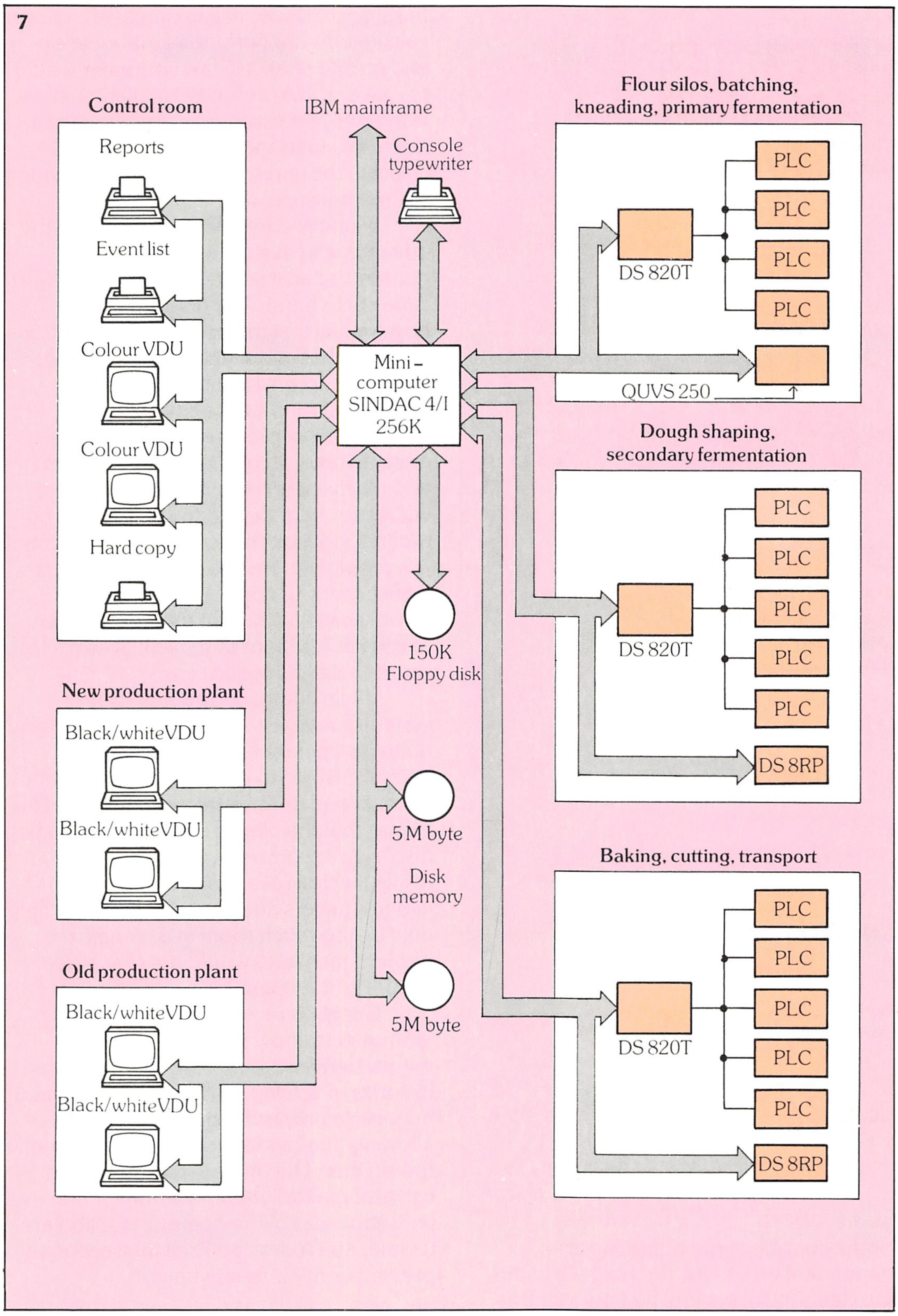
Complex processes often involve many kilometres of wiring just to transmit and receive data signals. The amount of wiring can be reduced, in some cases, by multiplexing, but this can cause problems if not properly managed. Microcomputers associated with control elements and sensors can also help in the management of communications to make it possible to share cable and reduce costs.

While computer process control systems are able to monitor processes, and optimise operations directly, plant operators need to observe and cope with unusual and unforeseen occurrences. This means that the plant's operation has to be displayed in some way. Conventional display techniques – indicator lamps, gauges, meters and the like – would simply take up too much room in a complex modern plant and would not give a clear picture of the operation.

Computer systems can be programmed to show selected measurements: derive summaries of data; provide histories of activity; and guide the operator by showing control sequences, but allowing him/her to proceed with or modify the actions. The process control system can automatically send warnings if the consequences of the operator's action are unsafe, and it can also itself take action to prevent or minimise damage.

The programmable logic controller

In a batch process, a sequence of actions has to be triggered by the controller. These actions could, for instance, simply involve turning valves or pumps on and off – as in



7. Structure of the control system for the Wasbröd crispbread plant, Sweden. *Source: ASEA Journal.*

milk processing for example. Here, vessels are filled with milk which is processed; the vessels are then emptied and charged with a sterilizing solution. This sterilizing solution is then drained off, steam is passed through the vessel and the cycle is repeated.

In some cases the sequence is timed, but subsequent steps in the process can

also be triggered by a detected condition in the process, such as a certain temperature being achieved, or a vessel filling. In either case, these simple on-off sequences can be controlled by a **programmable logic controller** (PLC), which is a microcomputer based device that utilises standard interfaces for sensors and powered control elements.

The microcomputer is programmed to work with the interfaces and to perform sequencing tasks. There is some flexiblity for users to adapt the program to suit their system, and this is often carried out in a simple graphical form. One set of notation that is used in these cases is the **relay ladder diagram**. This was originally devised for use by electricians who had to wire up pre-PLC electromechanical control systems. PLCs are dedicated computers and their simple task-related programming makes them an extremely flexible and adaptable control system component. Because of volume sales of PLCs and declining costs of microcomputers, the PLC can be employed at a lower cost than alternatives. The PLC is therefore enabling a greater number of processes to benefit from sophisticated control and interlock systems. An example of the structure of a control system is shown in *figure 7*.

Feedback control

The more elaborate forms of PLC allow feedback control, and are useful, for example, in the controlling of a sequence of regimes in an oven to heat treat a metal component. However, the performance of feedback controllers is often difficult to predict and in some cases instabilities can occur; that is to say, the quantity to be controlled undergoes wide uncontrolled fluctuations or moves towards an extreme in its operating range.

If control systems are to be operated with confidence, careful mathemetical and experimental analysis of their operation has to be carried out. This, of course, can be performed by a computer which is also able to simulate the industrial process and the proposed control systems. In this way, designers can push experimental systems to their limits, without the risk of injury or damage that might be incurred in a real plant.

CAD

Computer aided design (CAD) is a familiar and longstanding application of computers in industry. In the early days, computers aided scientists and engineers to more thoroughly analyse designs and scientific models because of the sheer amount of calculations that could be carried out. In this way, the need for some of the previous extensive, and expensive, experiments and detailed prototypes needed to confirm the validity of a design or theory was avoided.

In order to improve the usefulness of computers to design applications in manufacturing, two factors had to be overcome: first, work had to be done to express the engineers' description in a form that could be handled by the computer; and second, the cost.

Engineers and other designers have traditionally expressed their ideas as drawings, and gradually programs were developed that solved these problems and turned CAD into a graphics-based system.

The first commercially available computer aided draughting package was introduced by IBM in 1964. Applicon Inc. then marketed the first turnkey (complete) CAD system in 1970. As you can see from *figure 8*, it was not until 10 years later that CAD system usage really began to take off. This coincided with the general expansion of computer usage brought about by rapidly declining costs.

A computer based draughting workstation comprises: a screen to display the drawings; a means of indicating points on the screen, such as a light pen or graphics tablet; and a keyboard through which commands are given to the computer. The designer thus has an automated drawing system. Instead of the design being created on paper, it exists on a display screen. Once digitized in this way, the drawing can be altered, annotated and manipulated in many different ways.

Displays

To successfully emulate conventional engineering drawing on a CAD system, computer graphics systems had to be developed that had much better graphic capabilities than were conventionally

available. Much greater screen resolution was necessary because of the level of accuracy required and multicolour facilities were also needed.

Information is held about drawings in one of two ways, giving rise to two different types of graphics CRT – **vector graphics** and the **bit-mapped display**. In a vector graphics system, straight lines or vectors are drawn on the screen by means of storing co-ordinates. By storing in memory the two sets of co-ordinates necessary for each line, the line can then be reconstructed on the screen. Curved lines can be constructed from a series of short straight lines. The greater the resolution of the screen, the greater degree of accuracy obtained.

Bit-mapped displays, on the other hand, rely on information about each pixel (see *Communications* 4) being stored in memory as a unique code. Where the screen is single colour and single brightness, the bits in memory correspond only to whether the pixel is illuminated or not. The drawing on the screen is constructed from an array of illuminated dots which correspond to the pattern or **map** held in memory – hence the name bit-mapped.

More sophisticated versions of this system enable different colours and variations in intensity of the display. In this case more than one bit of memory is associated with each pixel to determine colour value and intensity. For a screen with, say, 256 possible colours and tones, eight bits are needed for each pixel. A typical draughting screen with an aspect ratio of 1:1 might have 1 million (1,000 × 1,000) pixels. A screenful of data therefore requires 8 million bits!

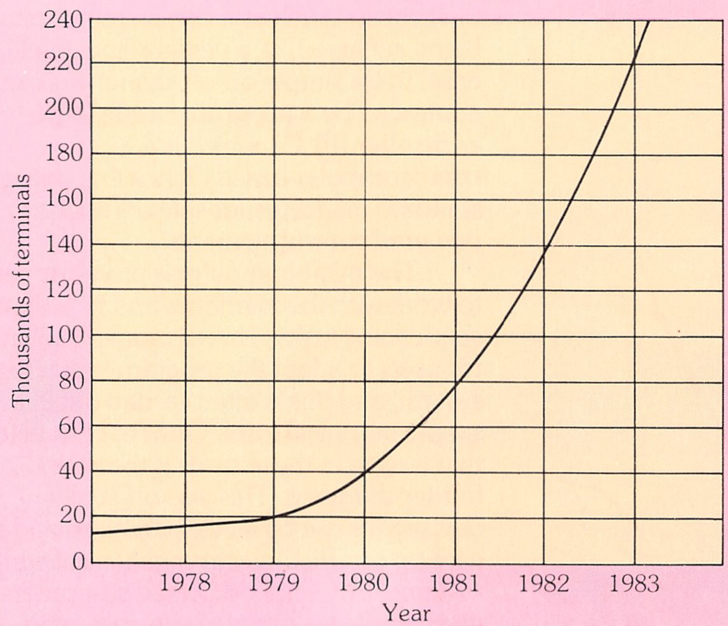
In order for the image to be maintained on the screen without the necessity for large amounts of memory, some vector graphics systems employ **storage tubes** that are able to retain the information before refreshing for up to several minutes. However, for those not employing such methods, the data must be continually refreshed by redrawing all the lines specified by memory in a **vector scan**.

Bit-mapped systems employ a **raster scan** technique to refresh the screen. (Raster scanning is discussed in detail in *Communications* 4.)

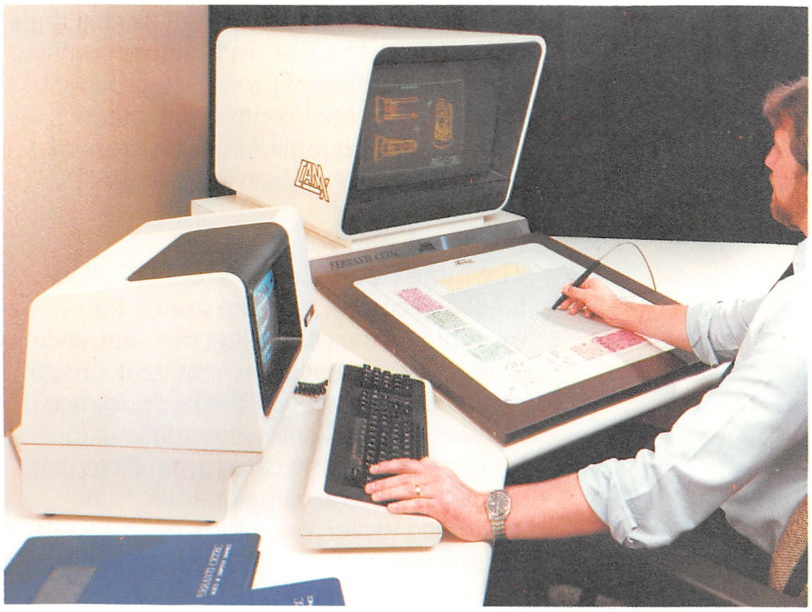
Once the 'design' is complete, hard copy output is usually required and is provided by means of a **plotter**. Single and multipen (corresponding to single and multicolour) plotters are available which are able to produce output indistinguishable from a hand draughted design.

(continued in part 39)

8



The boom in computer graphics



The Research House/Ferranti plc

8. CAD system usage began to mushroom in 1980.
Source: Voisinet, 'Introduction to CAD'.

Above: CAM-X computer aided design workstation.